



UNIVERSITÉ DE TECHNOLOGIE DE BRNO

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

FACULTÉ D'INGÉNIERIE MÉCANIQUE

FAKULTA STROJNÍHO INŽENÝRSTVÍ

INSTITUT DE MATHÉMATIQUES

ÚSTAV MATEMATIKY

APPLICATIONS DES QUATERNIONS EN ROBOT CINÉMATIQUE

APLIKACE QUATERNIONŮ V KINEMATICE ROBOTU

LA THÈSE DE MASTER

DIPLOMOVÁ PRÁCE

AUTEUR

AUTEUR PRÁCE

Docteur Diana

SUPERVISEUR

VEDOUČÍ PRÁCE

doc. Ing. Radomil Matousek, Ph.D.

BRNO 2019



Mémoire de mémoire de spécification

Département: Institut de Mathématiques
Étudiant: Docteur Diana
Programme d'études : Sciences appliquées en ingénierie
Champ d'étude: Ingénierie mathématique
Superviseur: doc. Ing. Radomil Matousek, Ph.D.
Année académique: 2018/19

Conformément à la loi n. 111/1998 concernant les universités et les règles d'études et d'exams BUT, vous ont été assignés le sujet suivant par la thèse de maîtrise du directeur de l'institut :

Applications des quaternions dans la cinématique des robots

Caractéristique concise de la tâche :

Cette thèse traite de l'utilité de la représentation quaternionique de la cinématique pour la robotique applications. La cinématique directe et inverse d'un manipulateur robotique est dérivée à l'aide de quaternions.

Objectifs Mémoire de Master :

L'examen des robots 6DOF et des modèles cinématiques directs et inverses pour la conception de contrôle.

Etat de l'art du contrôle de robots utilisant l'algèbre des quaternions.

Simulation de robot (Universal Robot) et application de programmation pratique.

Bibliographie recommandée :

SICILIANO, Bruno à KHATIB, Oussama, éd. Springer manuel de robotique. 2e édition. Berlin:

Springer, [2016]. ISBN 978-3-319-32550-7.

La date limite de soumission du mémoire de maîtrise est donnée par le calendrier de l'année académique 2018/19

À Brno,

LS

prof. RNDr. Josef Šlapal, CSc.
Directeur de l'Institut

doc. Ing. Jaroslav Katolický, Ph.D.
doyen FME

Abstrait

Cette thèse traite de l'utilité de l'application des quaternions dans la représentation de la cinématique des robots. Il commence par montrer la relation des quaternions avec les nombres complexes les plus connus et comment il peut représenter des rotations en trois dimensions. Ensuite, les quaternions doubles sont introduits pour représenter à la fois la rotation et la translation tridimensionnelles. Il sera ensuite utilisé pour dériver la cinématique avant et inverse, en particulier pour le robot universel UR3 qui est un bras robotique à 6 degrés de liberté. Enfin, une application réelle des quaternions doubles dans la programmation de robots sera démontrée.

Mots clés

quaternions, quaternions doubles, robotique, robot universel, UR3, cinématique directe, cinématique inverse, rotations tridimensionnelles

DOCTOR, DB Applications des Quaternions en Cinématique Robotique. Brno : Université de technologie de Brno, Faculté de génie mécanique, 2019. 105p.
Directeur de thèse de diplôme prof. doc. Ing. Radomil Matousek, Ph.D.

Je déclare avoir rédigé ce mémoire de diplôme Applications of Quaternions in Robot Kinematics par moi-même sous la direction de mon directeur de thèse, prof. doc. Ing. Radomil Matoušek, Ph.D., et l'aide d'Ing. Roman Parak, et en utilisant les sources listées dans la bibliographie.

31 mai 2019

Diana B. Docteur

Je voudrais d'abord dédier cette thèse à mon père, Mateo, qui m'a élevé comme la personne que je suis aujourd'hui, pleine d'espoirs et de rêves, et aux personnes derrière le programme InterMaths qui m'ont donné de multiples chances de réaliser ces rêves.

Diana B. Docteur

Contenu

1. Introduction	13
2 Nombres complexes 2.1	15
Histoire des nombres complexes	15
2.2 Le système de nombres complexes	16
2.2.1 Définition	16
2.2.2 Multiplication de nombres complexes	16
2.3 Représentation graphique de $z \in \mathbb{C}$	17
2.4 Propriétés des nombres complexes	18
2.4.1 Norme et Argument de $z \in \mathbb{C}$	18
2.4.2 Conjugué complexe de $z \in \mathbb{C}$	18
2.4.3 Règles algébriques pour les nombres complexes	19
2.4.4 Addition de nombres complexes	20
2.4.5 Multiplication de nombres complexes	20
2.5 Rotations dans le plan complexe	22
3 Quaternions 3.1	23
Histoire des Quaternions	23
3.2 Le système de numération des quaternions	26
3.2.1 Définition	26
3.2.2 Représentations des quaternions	26
3.2.3 Lois de multiplication des quaternions	27
3.3 Propriétés des quaternions	28
3.3.1 Conjugué de quaternion	28
3.3.2 Norme d'un quaternion	28
3.3.3 Quaternion unitaire	28
3.3.4 Inverse d'un quaternion	28
3.3.5 Addition et soustraction de quaternions	29
3.3.6 Multiplication et division de quaternions	29
3.3.7 Résumé des propriétés des quaternions	30
3.4 Rotations 3D à l'aide de quaternions	31

3.4.1 Représentations de rotation 3D	31
3.4.2 Conversions de quaternions	36
3.5 Quaternions doubles	38
3.5.1 Définition	38
3.5.2 Inverse d'un double quaternion	38
3.5.3 Multiplication d'un double quaternion	39
3.5.4 Transformation par un double quaternion	39
4 Application des quaternions au robot universel UR3	41
4.1 Robot universel - UR3	41
4.2 Cinématique directe par doubles quaternions	42
4.2.1 Représentations de la transformation des articulations	42
4.2.2 Calcul de la cinématique directe UR3	44
4.3 Cinématique inverse par doubles quaternions	45
4.3.1 Représentations des transformations d'articulations	45
4.3.2 Produits de transformation des joints	46
4.3.3 Transformations des cadres de la base aux poignets	47
4.3.4 Transformation des cadres de l'épaule au poignet	53
4.3.5 Solutions d'angles de joint	56
4.4 Application de programmation pratique	57
4.4.1 Paramètres Denavit-Hartenberg	57
4.4.2 Application de programmation Python	58
4.5 Vérification des résultats	71
5 Conclusions et Recommandations	75
5.1 Conclusion	75
5.2 Recommandation	77
Annexes	79
Un robot universel UR3	
Spécifications techniques	81
B Produits de transformation pour la base au poignet 3	85
C Produits de transformation pour l'épaule au poignet 1	97
D Liste des tableaux et figures	101

Chapitre 1

Introduction

Le concept de robotique a existé même pendant les temps anciens. Cependant, le terme «robot» n'a été conçu qu'au XXe siècle et est originaire de la République tchèque. Il a été introduit pour la première fois par les Tchèques

le dramaturge Karel Capek dans sa pièce "Rossum's Universal Robots (RUR)" dont la première a eu lieu à Prague dans les années 1920. Cependant, c'est le frère de Karel, Josef Capek, qui était vraiment l'inventeur du terme « robot » et il l'a dérivé du mot slave « robota » qui signifie travail subordonné ou forcé [2]. La pièce est rapidement devenue influente et le terme «robot» a ensuite été ajouté à la langue anglaise et a depuis été utilisé dans les fictions et dans les recherches scientifiques. À l'ère moderne, nous nous référons au robot comme une machine fabriquée par l'homme qui peut effectuer un travail et imiter les actions effectuées par les humains. Nous avons vu diverses applications de robots qui sont le plus souvent dans l'industrie manufacturière. Leur application la plus avancée serait les explorations marines et spatiales.

La modélisation d'un robot est l'endroit où la cinématique du robot entre en jeu. Formuler la bonne cinématique d'un robot est important pour analyser son comportement et le manipuler. Les robots sont composés d'articulations et de liens. Les deux types d'articulations les plus courants sont les articulations rotoïdes et prismatiques. Ces articulations effectuent respectivement des rotations et des translations. Nous pouvons comparer les articulations rotoïdes aux articulations humaines telles que les articulations de l'épaule, du coude, des poignets et des genoux. Un robot est également défini par ses degrés de liberté. Cela correspond à tous les mouvements possibles de ses articulations. Un robot planaire est décrit comme ayant trois degrés de liberté. C'est parce qu'il ne peut se déplacer que dans un avion. Ceux-ci sont à gauche ou à droite et en avant ou en arrière. L'autre degré de liberté est la rotation qui peut être effectuée dans un plan. Si un robot peut également effectuer un mouvement de haut en bas, qui est un axe de mouvement supplémentaire, il dispose désormais de quatre degrés de liberté. Si un robot peut aussi changer de pose (position

Chapitre 1 Introduction

et orientation) au moyen d'une rotation autour d'un autre axe dans le plan, puis il ajoute également à ses degrés de liberté. Ces rotations sont plus communément appelées tangage, lacet et roulis. Si un robot peut effectuer toutes ces translations et rotations, alors il a le maximum de six degrés de liberté.¹ Robot Kinematics est l'étude de tous

les mouvements possibles d'un robot sans tenir compte des forces qui provoquent ces mouvements comme indiqué dans [21] et [12].

La cinématique peut être divisée en cinématique directe et inverse. Forward Kinematics détermine la pose de l'effecteur terminal en fonction des angles d'articulation. Cela peut être facilement dérivé et la solution est toujours unique. Cependant, son complément, la cinématique inverse, est généralement beaucoup plus difficile à dériver car il détermine les angles d'articulation nécessaires pour une certaine pose. Les équations sont non linéaires et peuvent avoir des singularités. En dehors de cela, il peut exister également plusieurs solutions. Notre application implique la vérification de ces solutions ; cependant, la détermination de la solution optimale n'entre pas dans le cadre de l'étude.

La cinématique du robot est principalement décrite par la transformation effectuée par chacune des articulations du robot. Ces transformations sont décomposées en rotations et translations. Une articulation robotique ne peut effectuer qu'une translation telle que les articulations prismatiques, ce qui signifie qu'elle n'affecte pas l'orientation de l'effecteur terminal, mais uniquement sa position. Il existe également une articulation qui peut effectuer les deux, comme les articulations rotoïdes. Ces articulations modifient non seulement l'orientation de l'effecteur terminal mais peuvent également modifier sa position. Un exemple est si cette articulation est connectée à un lien avec l'effecteur terminal. Cette liaison constitue la translation entre ce cadre articulaire et le cadre effecteur. Ainsi, la rotation de l'articulation affectera également la longueur de la translation par rapport au cadre de coordonnées de base ainsi que l'orientation de l'effecteur terminal.

Pour les rotations, il existe différentes manières de les représenter. Les moyens les plus courants sont les matrices de rotation, les angles d'Euler, la représentation axe-angle et les quaternions unitaires. Cette étude porte sur les quaternions unitaires ce qui nous donne une représentation élégante et compacte de la rotation. Alors qu'une matrice de rotation 3×3 a besoin de 9 éléments pour représenter une rotation, les quaternions n'en ont besoin que de 4. Cependant, cette représentation n'est toujours pas autant utilisée que les transformations homogènes en robotique. C'est peut-être à cause du peu de connaissances que nous avons à ce sujet. Les quaternions sont une extension des nombres complexes. Ainsi, afin de saisir l'idée de base des quaternions, nous devons d'abord comprendre le concept de nombres complexes. Le but de cette recherche est de faciliter la compréhension de la relation entre les quaternions et les nombres complexes et de montrer comment les quaternions, plus particulièrement les quaternions duaux, sont capables de représenter au mieux les transformations dans l'espace tridimensionnel.

¹Extrait du cours edX : "Robotique : fondements cinématiques et mathématiques" dans [1]

Chapitre 2

Nombres complexes

Avant de commencer l'étude des quaternions, il nous faudra faire quelques rappels sur les idées de base et quelques propriétés importantes des nombres complexes qui sont très bénéfiques pour comprendre le concept de quaternions. Nous commencerons d'abord par expliquer la raison pour laquelle les nombres complexes ont été formulés, puis nous procéderons à la description de ce système de nombres. Ensuite, nous aborderons les propriétés importantes des nombres complexes en les définissant au sens algébrique et aussi au sens géométrique.¹

2.1 Histoire des nombres complexes

Le système de nombres réels $\mathbb{R}$ résulte de la recherche d'un système incluant les nombres rationnels et fournissant des solutions à diverses équations polynomiales. Cependant, il existe certaines équations polynomiales telles que $x^2 + 2x + 2 = 0$ qui ne peuvent être satisfaites par aucun nombre réel x . L'utilisation de la formule quadratique $\frac{-b \pm \sqrt{b^2 - 4ac}}{2a}$ donne des expressions formelles pour les deux solutions pour le système $ax^2 + bx + c = 0$; mais dans les cas des équations ci-dessus, cette formule implique des racines carrées de nombres négatifs qui n'étaient pas définis auparavant. Au XVI^e siècle, le mathématicien italien Geronimo Cardano a commencé à considérer ce type d'équations et a souligné la nécessité à la fois des nombres négatifs et des «nombres complexes». Il a remarqué que si ces expressions devaient être factorisées par $\sqrt{-1}$ avec la propriété que $\sqrt{-1} \cdot \sqrt{-1} = -1$, elles satisfaisaient en effet comme solutions à ces équations. Nous utilisons maintenant la désignation largement acceptée $i = \sqrt{-1}$ pour ces expressions telles que $i^2 = -1$.

¹Le contenu de ce chapitre, y compris les images, est tous référencé dans le livre "Basic Complex Analysis" de JE Marsden et MJ Hoffman [18]

2.2 Le système de nombres complexes

Nous commençons par définir un nombre complexe et expliquer la règle de multiplication des nombres complexes qui est nécessaire pour comprendre la multiplication des quaternions.

2.2.1 Définition

Le système de nombres complexes, noté $\mathbb{C}$, est l'ensemble $\mathbb{R}^2$ le plan xy lequel est constitué de toutes les paires ordonnées (x, y) de nombres réels avec les règles habituelles d'addition vectorielle,

$$(x_1, y_1) + (x_2, y_2) = (x_1 + x_2, y_1 + y_2) \quad (2.1)$$

et multiplication scalaire par un nombre réel a ,

$$a(x, y) = (ax, ay) \quad (2.2)$$

et avec l'opération supplémentaire de multiplication complexe définie par,

$$(x_1, y_1)(x_2, y_2) = (x_1x_2 - y_1y_2, x_1y_2 + y_1x_2) \quad (2.3)$$

Pour la représentation des nombres complexes, nous identifions d'abord les nombres réels x comme des points sur l'axe réel x et y comme les points sur l'axe imaginaire y où le point unitaire $(0, 1)$ représente i . Alors maintenant, nous pouvons représenter un nombre complexe comme $(x, y) = x + iy$ qui est la notation la plus standard. Ainsi, un seul symbole tel que $z = a + ib$ peut également être utilisé pour indiquer un nombre complexe où $\text{Re} z = a$ est appelé la partie réelle et $\text{Im} z = b$ est appelé la partie imaginaire. Par conséquent, un nombre réel est un nombre complexe z où $\text{Im} z = b = 0$.

Si nous avons le contraire, $\text{Re} z = a = 0$, alors nous avons un nombre imaginaire pur. La notation $z \in \mathbb{C}$ implique que z appartient à l'ensemble des nombres complexes.

2.2.2 Multiplication de nombres complexes

Pour dériver la règle de multiplication des nombres complexes,

$$(a, b)(c, d) = (ac - bd, ad + bc) \quad (2.4)$$

nous utilisons les notations standard pour les nombres complexes (a, b) et (c, d) , puis en appliquant la propriété $i^2 = -1$, nous obtenons :

$$\begin{aligned} (a + ib)(c + id) &= ac + iad + ibc + i^2bd \\ &= ac + i(ad + bc) + (-1)bd = ac - bd + i(ad + bc) \end{aligned} \quad (2.5)$$

2.3 Représentation graphique de $z \in \mathbb{C}$

Comme nous l'avons indiqué dans la définition, un nombre complexe peut également être considéré comme un vecteur bidimensionnel dans $\mathbb{R}^2$ de l'origine à un point du plan xy avec des coordonnées données par les parties réelle et imaginaire du nombre complexe. Voir Figure 4.6.

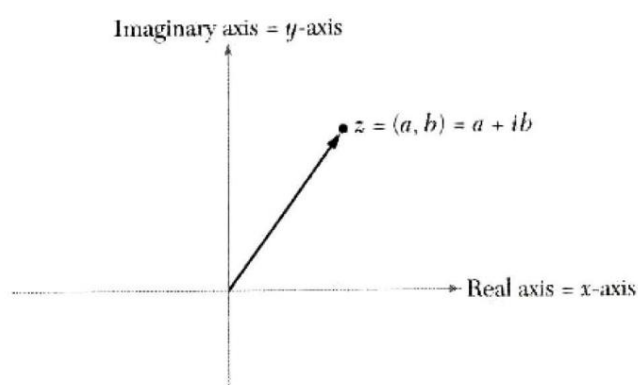


Figure 2.1 : Représentation vectorielle de $z = a + ib$ avec $\operatorname{Re} z = a$ et $\operatorname{Im} z = b$.

Coordonnées polaires et représentation exponentielle. Nous écrivons un nombre complexe sous la forme de coordonnées polaires en prenant la longueur r du vecteur $(a, b) = a + ib$ par $r = \sqrt{a^2 + b^2}$. Ensuite, on suppose que le vecteur fait un angle positif θ avec l'axe réel où $0 \leq \theta < 2\pi$. Voir Figure 2.2.

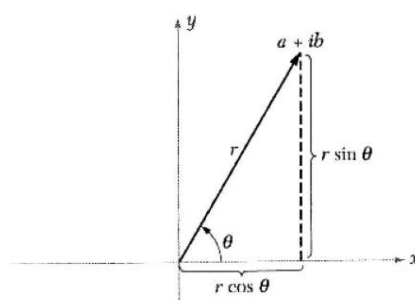


Figure 2.2 : Représentation en coordonnées polaires du nombre complexe $z = a + ib$ de longueur r et d'angle θ .

Par SOHCAHTOA, on obtient que $\tan \theta = \frac{b}{a}$ où $a = r \cos \theta$ et $b = r \sin \theta$, donc on a,

$$z = a + ib = r \cos \theta + i(r \sin \theta) = r(\cos \theta + i \sin \theta) = re^{i\theta} \quad (2.6)$$

²Nous avons dérivé la forme exponentielle en utilisant la formule d'Euler, $e^{i\theta} = \cos \theta + i \sin \theta$.

2.4 Propriétés des nombres complexes

Dans cette section, nous présenterons toutes les propriétés fondamentales et importantes des nombres complexes nécessaires à la compréhension des propriétés correspondantes des quaternions.

2.4.1 Norme et Argument de $z \in \mathbb{C}$

Dans la figure 2.2, le vecteur représentant le nombre complexe $z = a + ib$ a une longueur r qui peut également être notée $|z|$ et est appelée la norme de z . Le calcul formel est donné par :

$$|z| = \sqrt{a^2 + b^2} \quad (2.7)$$

L'angle θ est appelé l'argument de z et est noté $\arg z = \theta$. Le calcul formel est donné par :

$$\theta = \tan^{-1} \frac{b}{a} \quad (2.8)$$

2.4.2 Conjugué complexe de $z \in \mathbb{C}$

Étant donné un nombre complexe $z = a + ib$, alors $\bar{z}$, qui est appelé le conjugué complexe de z , est défini comme $\bar{z} = a - ib$. Cela peut être représenté géométriquement comme une réflexion dans l'axe réel x illustré à la figure 2.3.

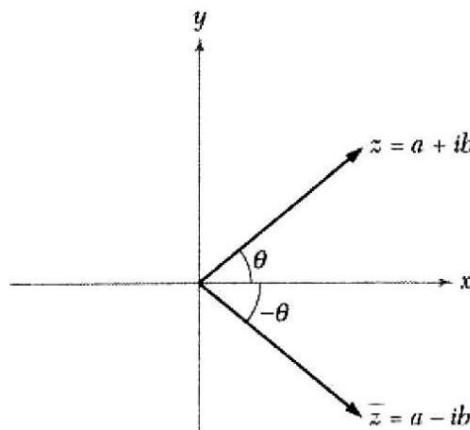


Figure 2.3 : Conjugaison complexe.

Par conséquent, sous forme de coordonnées polaires, le conjugué complexe peut également être représenté en utilisant la même amplitude mais un angle opposé.

2.4.3 Règles algébriques pour les nombres complexes

Dans cette section, nous montrerons que toutes les règles algébriques usuelles dans la manipulation des nombres réels peuvent également s'appliquer aux nombres complexes.

Règles d'ajout

Les nombres complexes z et w satisfont aux règles d'addition suivantes :

1. $z + w = w + z$
2. $z + (w + s) = (z + w) + s$
3. $z + 0 = z$
4. $z + (-z) = 0$

Règles de multiplication

Les nombres complexes z et w satisfont aux règles de multiplication suivantes :

1. $zw = wz$
2. $(zw)s = z(ws)$
3. $1z = z$
4. $z(z^{-1}) = 1$ pour $z \neq 0$

Nous pouvons résoudre l'inverse multiplicatif z^{-1} une $z = a + ib$ par :

$$z^{-1} = \frac{a}{a^2 + b^2} \quad \text{et} \quad b^{-1} = \frac{-b}{a^2 + b^2} \quad (2.9)$$

Donc, pour $z = a + ib \neq 0$, on peut écrire :

$$z^{-1} = \frac{a}{a^2 + b^2} - \frac{ib}{a^2 + b^2} \quad (2.10)$$

Loi distributive

Les nombres complexes z , w et s satisfont les règles de distribution suivantes,

$$z(w + s) = zw + zs \quad (2.11)$$

Il est important de noter ici que, tout comme les nombres réels, les nombres complexes satisfont également les propriétés associatives, commutatives et distributives de l'addition et de la multiplication.

Nous comparerons ce cas au cas des opérations quaternioniques qui sera traité dans le chapitre suivant.

2.4.4 Addition de nombres complexes

Nous pouvons illustrer l'addition de nombres complexes comme l'addition de vecteurs bidimensionnels représentant les deux nombres complexes. Cette opération est illustrée à la figure 2.4.

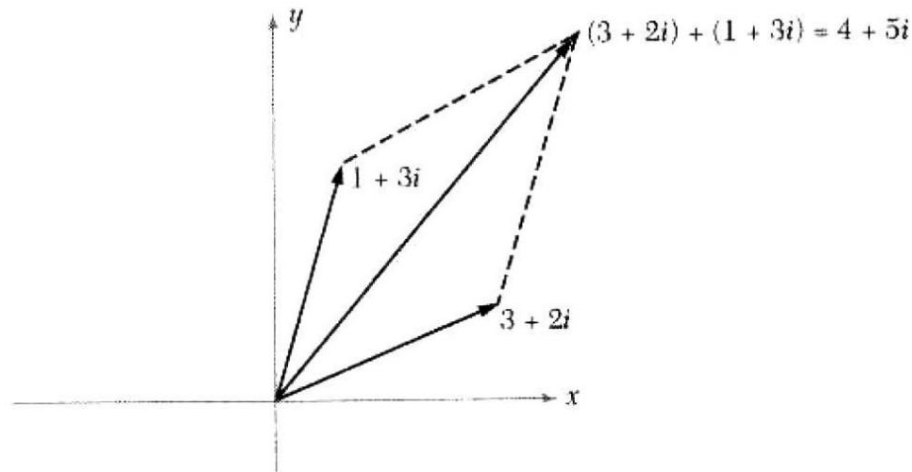


Figure 2.4 : Addition des nombres complexes $3 + 2i$ et $1 + 3i$.

La même représentation peut également être utilisée pour la soustraction de nombres complexes où l'addition peut également être utilisée en ajoutant le premier vecteur à la réflexion à l'origine de l'autre vecteur.

2.4.5 Multiplication de nombres complexes

Représenter un nombre complexe en coordonnées polaires ou sous forme exponentielle aide à comprendre le produit de deux nombres complexes au sens géométrique.

Si nous posons $z_1 = r_1(\cos \theta_1 + i \sin \theta_1)$ et $z_2 = r_2(\cos \theta_2 + i \sin \theta_2)$, nous avons,

$$\begin{aligned} z_1 z_2 &= r_1(\cos \theta_1 + i \sin \theta_1) \cdot r_2(\cos \theta_2 + i \sin \theta_2) = \\ &= r_1 r_2 [(\cos \theta_1 \cdot \cos \theta_2 - \sin \theta_1 \cdot \sin \theta_2) + i(\cos \theta_1 \cdot \sin \theta_2 + \cos \theta_2 \cdot \sin \theta_1)] \\ &= r_1 r_2 [\cos(\theta_1 + \theta_2) + i \sin(\theta_1 + \theta_2)] = r_1 r_2 e^{i(\theta_1 + \theta_2)} \end{aligned} \quad (2.12)$$

Notez que nous avons appliqué les formules trigonométriques suivantes pour la somme et la différence des produits des cosinus et des sinus des angles :

$$\cos(\theta_1 + \theta_2) = \cos \theta_1 \cos \theta_2 - \sin \theta_1 \sin \theta_2 \quad (2.13)$$

$$\sin(\theta_1 + \theta_2) = \cos \theta_1 \sin \theta_2 + \cos \theta_2 \sin \theta_1 \quad (2.14)$$

2.4. Propriétés des nombres complexes

Proposition. Pour tout nombre complexe z_1 et z_2 , nous proposons que,

$$|z_1 z_2| = |z_1| \cdot |z_2| \text{ et } \arg(z_1 z_2) = \arg z_1 + \arg z_2 \pmod{2\pi} \quad (2.15)$$

Cela signifie que le produit de deux nombres complexes est un nombre complexe qui a une longueur égale aux produits des longueurs des deux nombres complexes qui ont été multipliés et qui a un argument égal à la somme des arguments de ces deux nombres complexes. Ce produit peut être illustré dans la Figure 2.5 et un exemple dans la 2.6.

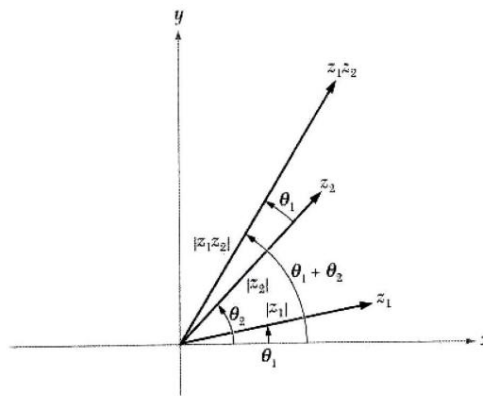


Figure 2.5 : Multiplication des nombres complexes z_1 et z_2 .

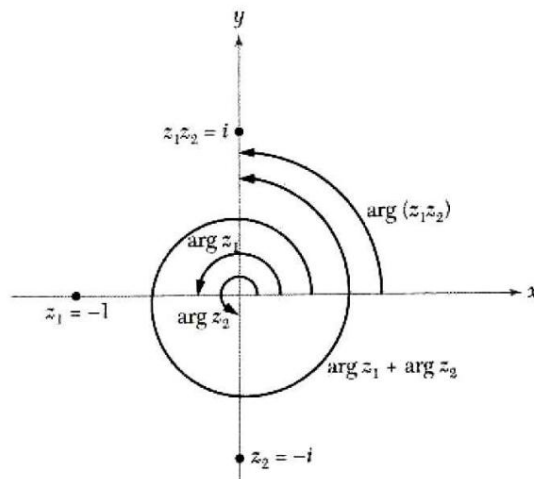


Figure 2.6 : Multiplication de nombres complexes -1 et $-i$.

2.5 Rotations dans le plan complexe

La multiplication des nombres complexes peut aussi être analysée par une application ψ_z qui représente la multiplication par z telle que $\psi_z : \mathbb{C} \rightarrow \mathbb{C}$ et est définie par $\psi_z(w) = wz$. Par la proposition, l'effet de cette application est de faire tourner un nombre complexe w d'un angle égal à $\arg z$ dans le sens inverse des aiguilles d'une montre et d'étirer sa longueur du facteur $|z|$. Cela peut être vu dans la Figure 2.7 et également dans l'exemple précédent de la Figure 2.6.

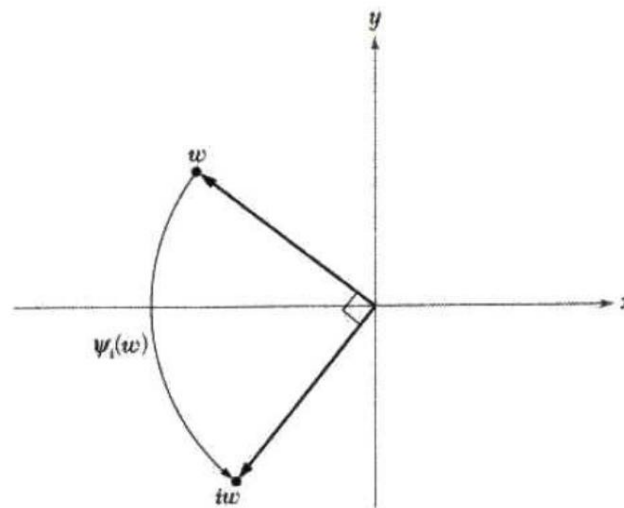


Figure 2.7 : Multiplication de wz où $z = i$.

Dans le chapitre suivant, nous commencerons par la discussion des quaternions et comment sa multiplication peut également représenter des rotations en trois dimensions.

chapitre 3

Quaternions

Ce chapitre traite de la discussion principale des quaternions et de ses propriétés de base. Nous commençons par détailler la découverte spectaculaire des quaternions et comment il a ébranlé le monde des mathématiques en son temps. Ensuite, nous poursuivrons avec la définition et les propriétés importantes des quaternions qui nous donnera une compréhension de base de ce système de numération. Il est également important montrer comment les quaternions peuvent représenter des rotations en trois dimensions et comment il se rapporte à d'autres représentations de rotation. Enfin, nous discuterons sur les quaternions doubles et leur utilisation en robotique.

3.1 Histoire des quaternions

Comme nous l'avons vu dans le chapitre précédent, les nombres complexes modélisent l'espace à deux dimensions en ayant une partie réelle et imaginaire qui correspond au plan xy dans lequel la multiplication et l'addition de ces nombres surviennent dans le même espace. Pendant des années, le mathématicien irlandais William Rowan Hamilton avait essayé de modéliser un espace tridimensionnel qui a ce même structuration. Il essayait de le représenter avec un réel et à deux dimensions partie imaginaire mais cela ne satisfaisait pas la structure qu'il voulait réaliser. Son fils, William Edward Hamilton, lui posait souvent des questions sur ses progrès dans la multiplication des triplets auxquels il répondrait qu'il ne peut qu'ajouter ou soustrayez-les mais ne les multipliez jamais. [24]

L'acte de vandalisme mathématique le plus célèbre. La découverte arriva le 16 octobre 1843 alors qu'Hamilton marchait le long de la Brougham Bridge (aujourd'hui Broom Bridge) à Dublin. Il a eu un éclair de perspicacité et n'a pas pu s'empêcher de graver dans le pont les équations suivantes :

$$i^2 = j^2 = k^2 = ijk = -1 \quad (3.1)$$

Chapitre 3. Quaternions

Dans son journal [10], Hamilton a détaillé ce moment avec l'écriture suivante :

"Le 16 octobre - qui se trouvait être un lundi et un jour du Conseil de la Royal Irish Academy - j'entrais pour assister à une présidence, et votre mère marchait avec moi, le long du canal royal, auquel elle avait peut-être conduit; et bien qu'elle me parlât de temps à autre, un sous-courant de pensée se poursuivait dans mon esprit, qui donna enfin un résultat, dont il n'est pas exagéré de dire que je sentis tout de suite l'importance. Un circuit électrique semblait se fermer ; et une étincelle jaillit, annonciatrice (comme je le prévoyais immédiatement) de longues années à venir d'une pensée et d'un travail définitivement dirigés, par moi-même si j'en étais épargné, et en tout cas de la part des autres, si j'étais même autorisé à le faire. vivre assez longtemps distinctement pour communiquer la découverte. Je n'ai pas non plus pu résister à l'impulsion - aussi peu philosophique soit-elle - de graver au couteau sur une pierre de Brougham Bridge, au passage, la formule fondamentale avec les symboles i, j, k ; à savoir, $i^2 = j^2 = k^2 = ijk = -1$ qui contient la solution du problème.

$$i^2 = j^2 = k^2 = ijk = -1$$

Sa découverte de ces lois de multiplication a également été un moment décisif dans l'histoire de l'algèbre. À ce moment, il réalisa qu'une quatrième dimension était nécessaire et il inventa plus tard le terme quaternion pour représenter l'espace réel couvert par les éléments $1, i, j$ et k soumis aux lois de multiplication ci-dessus.

Ses sculptures se sont usées avec le temps mais une plaque a été placée dans le pont pour le remplacer et pour commémorer la découverte dans l'histoire mathématique.



Figure 3.1 : Un portrait de Sir William Rowan Hamilton¹

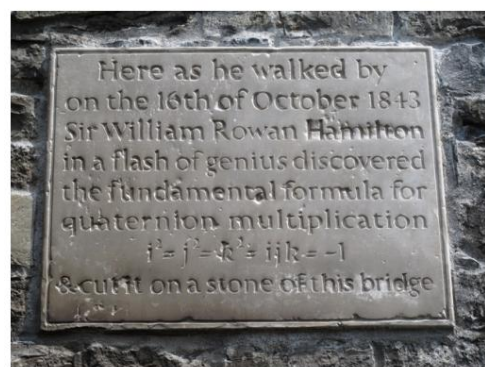


Figure 3.2 : Plaque commémorative de la découverte des quaternions.²

3.1. Histoire des quaternions

12 Selon les mots exacts de Hamilton [10],

« Un quaternion peut être symboliquement défini comme étant une expression quadrinomiale de la forme, $q = w + ix + jy + kz$ dans laquelle w, x, y, z sont quatre scalaires ou quantités algébriques ordinaires, tandis que i, j et k sont trois nouveaux symboles obéissant à toutes les lois contenues dans la formule ci-dessus et donc non soumis à toutes les règles usuelles de l'algèbre.

Hamilton a défini les quaternions comme un quotient de deux vecteurs. Les quaternions étant un système de nombres qui ne satisfont pas à la règle commutative habituelle de multiplication, ils ont inévitablement rencontré une certaine résistance dans le monde mathématique lorsqu'il a été découvert et expliqué pour la première fois par Hamilton. La rivalité entre ceux qui préfèrent les quaternions et ceux qui préfèrent les vecteurs comme notations physiques s'est transformée en une guerre entre ceux qui se sont rangés du côté des quaternions et ceux qui se sont rangés du côté de la notation vectorielle, certains faisant même référence aux quaternions comme un mal sans mélange.

Avec l'introduction des quaternions, elle a ouvert de nouvelles possibilités algébriques comme la redécouverte des octaves ou octonions, une algèbre en huitième dimension, de John T. Graves réalisée en généralisant avec succès les quaternions. Un autre résultat bien connu des quaternions est ses généralisations à des dimensions supérieures par William Clifford qui a développé un moyen de construire des algèbres à partir de formes quadratiques de sorte que les quaternions de Hamilton apparaissent comme cette algèbre plus connue sous le nom d'algèbre de Clifford.

À partir du milieu des années 1880, les quaternions ont commencé à être remplacés par l'analyse vectorielle qui décrivait les mêmes phénomènes que les quaternions mais conceptuellement plus simples et plus propres sur le plan de la notation. Cependant, les quaternions ont connu un renouveau depuis la fin du 20^e siècle en raison de leur utilité pour décrire les rotations spatiales. En effet, les quaternions offrent une manière élégante de décrire et de calculer les rotations qui est compacte et plus efficace en termes de calcul et évite le blocage de Cardan qui se produit lorsqu'un degré de liberté est perdu lorsque certaines rotations sont appliquées. Euler Angles, en particulier, est sensible à ce problème mécanique.

Pour ces raisons, les quaternions sont encore utilisés dans l'infographie, la vision par ordinateur, la robotique, la théorie du contrôle, le traitement du signal, le contrôle d'attitude, la mécanique quantique et de nombreux autres domaines qui impliquent de travailler dans un espace tridimensionnel.

¹https://en.wikipedia.org/wiki/William_Rowan_Hamilton

²<http://www.irelandtravelkit.com/broome-bridge-and-sir-hamiltons-eureka-moment-cabra-co-dublin/broombridge-plaque-bdolan-2010-800-label/>

3.2 Le système de numération des quaternions

Nous commençons par la définition du quaternion et ses lois de multiplication étendues comme également indiqué dans [13].

3.2.1 Définition

Les quaternions sont l'ensemble défini comme,

$$H = \{w + ix + jy + kz \mid w, x, y, z \in \mathbb{R}\} \quad (3.2)$$

qui est l'ensemble constitué de tous les nombres pouvant être sous la forme,

$$q = w + ix + jy + kz \text{ où } w, \quad (3.3)$$

x, y et z sont des nombres réels et i, j et k sont ce que nous appelons les unités quaternion fondamentales qui représentent les trois dimensions imaginaires.

3.2.2 Représentations des quaternions

Formulaire Quaternaire Standard. Un quaternion est généralement représenté en utilisant sa forme quaternionnaire standard écrite comme,

$$q = q_0 + q_1i + q_2j + q_3k \quad (3.4)$$

dans laquelle toutes les variables correspondent aux mêmes interprétations que dans la définition.

Forme scalaire et vectorielle. Outre la forme standard, un quaternion peut également être décrit comme composé d'une partie scalaire et d'une partie vectorielle où, dans le cas ci-dessus, la partie quaternion, $q_1i + q_2j + q_3k$, représente la partie vectorielle et q_0 la partie scalaire. Cela peut être représenté sous la forme dite de vecteur scalaire écrit comme,

$$q = (q_0, \mathbf{q}) \quad (3.5)$$

où q_0 est appelé la partie scalaire et le vecteur $\mathbf{q}$ représente le vecteur tridimensionnel formé par la partie imaginaire du quaternion et est appelé la partie vectorielle.

Un quaternion sans partie réelle est appelé quaternion pur ou quaternion vectoriel et un quaternion sans partie imaginaire est alors équivalent à son nombre réel correspondant, ce qui implique que les nombres réels et complexes sont un sous-ensemble des quaternions. ($\mathbb{R} \subset \mathbb{C} \subset H$)

3.2.3 Lois de multiplication des quaternions

Ces unités quaternion i , j et k , comme indiqué précédemment, satisfont aux conditions suivantes règle de multiplication :

$$i^2 = j^2 = k^2 = -1 \quad (3.6)$$

qui est étendu aux lois de multiplication suivantes :

$$ij = k = -ji \quad (3.7)$$

$$jk = i = -kj \quad (3.8)$$

$$ki = j = -ik \quad (3.9)$$

Ces lois sont résumées dans le tableau 4.1.

$\times$	1	i	j	k
1	1	i	j	k
i	-1	k	$-j$	
j	$-k$	-1	i	
k	i	j	-1	

Tableau 3.1 : Tableau de multiplication des quaternions

Ceux-ci peuvent également être visualisés dans le diagramme illustré à la Figure 3.3.³

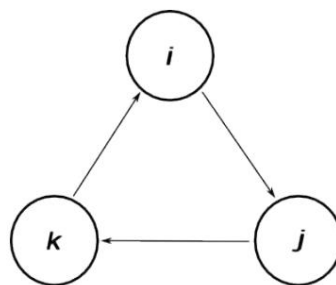


Figure 3.3 : Diagramme de multiplication de quaternions

Comme on peut le voir sur le tableau et le schéma ci-dessus, la multiplication des unités quaternion ne sont pas commutatives ($ij \neq ji$). Si la multiplication de ces unités vont à contre-courant dans le diagramme, le produit sera le négatif de le produit correspondant dans le flux normal. Cela se comprendra plus clairement une fois que nous avons discuté de la représentation des rotations par des quaternions.

³<http://hforsten.com/quaternion-roots-of-modern-vector-calculus.html>

3.3 Propriétés des quaternions

Dans cette section, nous présenterons toutes les propriétés de base et importantes d'un quaternion donné comme $q = q_0 + q_1i + q_2j + q_3k$ comme également indiqué dans [17].

3.3.1 Conjugué de quaternions

Le conjugué quaternion de q est donné en annulant sa partie vectorielle représentée par,

$$q^* = q_0 - q_1i - q_2j - q_3k = (q_0, -q) \quad (3.10)$$

3.3.2 Norme d'un quaternion

La norme du quaternion, également appelée magnitude, est la longueur du quaternion à partir de l'origine. Il est défini par :

$$\|q\| = \sqrt{qq^*} = \sqrt{q_0^2 + q_1^2 + q_2^2 + q_3^2} \quad (3.11)$$

3.3.3 Quaternion unitaire

Un quaternion est appelé quaternion unitaire s'il a une norme de 1, c'est-à-dire

$$\|q\| = \sqrt{q_0^2 + q_1^2 + q_2^2 + q_3^2} = 1 \quad (3.12)$$

Normalisation. Tout quaternion peut être normalisé pour le transformer en un quaternion unitaire. Cela se fait en divisant le quaternion par sa grandeur ou sa norme, par exemple :

$$\hat{q} = \frac{q}{\|q\|} = \frac{q_0 + q_1i + q_2j + q_3k}{\sqrt{q_0^2 + q_1^2 + q_2^2 + q_3^2}} \quad (3.13)$$

3.3.4 Inverse d'un quaternion

L'inverse (réciproque) d'un quaternion peut être calculé par,

$$q^{-1} = \frac{q^*}{\|q\|^2} = \frac{q_0 - q_1i - q_2j - q_3k}{q_0^2 + q_1^2 + q_2^2 + q_3^2} \quad (3.14)$$

Ceci peut être vérifié en utilisant la définition de la norme, ($\|q\|^2 = qq^*$),

$$qq^{-1} = q \frac{q^*}{\|q\|^2} = \frac{qq^*}{\|q\|^2} = 1 \quad (3.15)$$

3.3.5 Addition et soustraction de quaternions

L'addition des quaternions suit la même règle que dans les nombres complexes qui se fait par composant. Si nous considérons le quaternion donné q et un autre quaternion p donné comme,

$$p = p_0 + p_1i + p_2j + p_3k \quad (3.16)$$

Alors leur somme est donnée par,

$$p + q = (p_0 + q_0) + (p_1 + q_1)i + (p_2 + q_2)j + (p_3 + q_3)k \quad (3.17)$$

La même règle est également appliquée à la soustraction de quaternions tels que, (3.18)

$$p - q = (p_0 - q_0) + (p_1 - q_1)i + (p_2 - q_2)j + (p_3 - q_3)k$$

3.3.6 Multiplication et division de quaternions

Le produit de deux quaternions doit satisfaire aux lois fondamentales énoncées au début ainsi qu'à sa propriété de non-commutativité pour la multiplication. Ainsi, le produit des quaternions p et q est donné par,

$$\begin{aligned} pq &= (p_0 + p_1i + p_2j + p_3k)(q_0 + q_1i + q_2j + q_3k) \quad (3.19) \\ &= p_0q_0 + (p_0q_1 + p_1q_0)i + (p_0q_2 + p_2q_0)j + (p_0q_3 + p_3q_0)k \\ &\quad + (p_1q_2)ij + (p_2q_1)ji + (p_2q_3)jk + (p_3q_2)kj + (p_3q_1)ki + (p_1q_3)ik + (p_1q_1)i = \\ &\quad p_0q_0 + \quad^2 + (p_2q_2)j \quad^2 + (p_3q_3)k \quad^2 \\ &\quad (p_0q_1 + p_1q_0)i + (p_0q_2 + p_2q_0)j + (p_0q_3 + p_3q_0)k + (p_1q_2)(k) + \\ &\quad (p_2q_1)(-k) + (p_2q_3)(i) + (p_3q_2)(-i) + (p_3q_1)(j) + (p_1q_3)(-j) + (p_1q_1)(-1) + (p_2q_2)(-1) + \\ &\quad (p_3q_3)(-1) = p_0q_0 - p_1q_1 - p_2q_2 - p_3q_3 + (p_0q_1 \\ &\quad + p_1q_0 + p_2q_3 - p_3q_2)i \\ &\quad + (p_0q_2 + p_2q_0 + p_3q_1 - p_1q_3)j + (p_0q_3 + p_3q_0 + p_1q_2 - p_2q_1)k \end{aligned}$$

C'est une façon compliquée de prendre le produit de deux quaternions. Il existe un moyen beaucoup plus simple qui consiste à utiliser la forme scalaire et vectorielle d'un quaternion et à appliquer le produit scalaire et croisé pour les vecteurs. Ceci est fait par ce qui suit, $pq = (p_0, p)(q_0, q) = p_0q_0$

$$- p \cdot q + p_0q + q_0p + p \times q \quad (3.20)$$

Pour la division entre quaternions, cela peut être fait en prenant l'inverse multiplicatif du diviseur tel que,

$$\frac{p}{q} = pq^{-1} \quad (3.21)$$

et appliquer la méthode ci-dessus pour la multiplication de deux quaternions.

4 Cette addition et multiplication de quaternions est référencée dans [6]

Chapitre 3. Quaternions

3.3.7 Propriétés résumées des quaternions

Voici les propriétés résumées des quaternions.

Propriétés supplémentaires

Les quaternions q , p et r satisfont aux propriétés d'addition suivantes :

- Fermeture : $p + q \in \mathbb{H}$
- Associativité : $p + (q + r) = (p + q) + r$
- Commutativité : $p + q = q + p$
- Identité : $p + 0 = 0 + p = p$
- Somme : $p + q = (p_0 + q_0) + (p_1 + q_1)i + (p_2 + q_2)j + (p_3 + q_3)k$
- Différence : $p - q = (p_0 - q_0) + (p_1 - q_1)i + (p_2 - q_2)j + (p_3 - q_3)k$

Propriétés de multiplication

Les quaternions q , p et r satisfont aux propriétés de multiplication suivantes :

- Fermeture : $pq \in \mathbb{H}$
- Associativité : $(pq)r = p(qr)$
- Non-commutativité : $pq \neq qp$
- Distributivité : $p(q + r) = pq + pr$
- Inverse : Si $q \neq 0$ alors $q^{-1} \in \mathbb{H}$ tel que $qq^{-1} = q^{-1}q = 1$
- Produit : $pq = p_0q_0 - p \cdot q + p_0q + q_0p + p \times q$
- Division : $p/q = pq^{-1}$

Conjugué : $\bar{q} = q_0 - q_1i - q_2j - q_3k$

Norme : $\|q\| = \sqrt{q_0^2 + q_1^2 + q_2^2 + q_3^2}$

Quaternion unitaire : $\|q\| = 1$ et $p = 0, \|p\| = 1$ est un quaternion unitaire

Inverse : $q^{-1} = \bar{q}/\|q\|^2$ pour un quaternion unitaire u , $u^{-1} = u$

Produit scalaire : $q \cdot p = q_0p_0 + q_1p_1 + q_2p_2 + q_3p_3$

3.4 Rotations 3D à l'aide de quaternions

Dans cette section, nous montrerons comment les quaternions unitaires sont capables de représenter des rotations en trois dimensions et sa relation avec d'autres représentations de rotation. ⁵

3.4.1 Représentations de rotation 3D

Le concept d'expression d'une rotation en trois dimensions sous forme de formulation mathématique s'étend à la mécanique classique où la cinématique de rotation est la science décrivant le mouvement purement rotatif d'un objet avec des nombres.

Selon le théorème de rotation d'Euler, un déplacement général d'un corps rigide avec un point fixe est décrit par une rotation autour d'un axe. De plus, cette rotation peut être décrite de manière unique par un minimum de trois paramètres. Il existe différentes manières de représenter les rotations 3D, certaines nécessitant plus que trois paramètres, même si elles n'ont toutes que trois degrés de liberté. Les représentations de rotation les plus utilisées sont détaillées ici et sont chacune discutées.

Représentation de la matrice de rotation

Nous savons que les rotations bidimensionnelles peuvent être représentées par la matrice,

$$R = \begin{pmatrix} \cos \theta & -\sin \theta \\ \sin \theta & \cos \theta \end{pmatrix} \quad (3.22)$$

où θ est l'angle de rotation autour de l'origine. Avoir un seul paramètre d'angle implique également que les rotations planes n'ont qu'un seul degré de liberté.

En trois dimensions, on a affaire à des rotations sur les axes principaux x , y et z . Une rotation autour de l'axe x correspond à une rotation dans le plan yz .

D'autre part, une rotation autour de l'axe y correspond à une rotation dans le plan xz et une rotation autour de l'axe z correspond à une rotation dans le plan xy .

Ces rotations sont représentées par les matrices suivantes,

$$R_x = \begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos \theta & -\sin \theta \\ 0 & \sin \theta & \cos \theta \end{pmatrix} \quad (3.23)$$

⁵Cette sous-section est une compilation du contenu des sources suivantes : [5][6][8][9][17]

Chapitre 3. Quaternions

$$R_y = \begin{pmatrix} \cos \theta & 0 & \sin \theta \\ 0 & 1 & 0 \\ 0 & -\sin \theta & \cos \theta \end{pmatrix} \quad (3.24)$$

$$R_z = \begin{pmatrix} \cos \theta & -\sin \theta & 0 \\ \sin \theta & \cos \theta & 0 \\ 0 & 0 & 1 \end{pmatrix} \quad (3.25)$$

En général, si nous considérons un corps rigide avec trois vecteurs unitaires orthogonaux qui lui sont fixés (également appelés base). La spécification des coordonnées de cette base dans sa position de rotation actuelle en termes d'axes de coordonnées de référence sans rotation décrira la rotation qui lui est appliquée. Ces trois vecteurs unitaires r_1 , r_2 et r_3 , composés de 3 coordonnées chacun, donnent un total de 9 paramètres. Ces paramètres forment une matrice 3x3 appelée matrice de rotation.

Les éléments d'une matrice de rotation sont définis comme suit :

$$R = [r_1 \ r_2 \ r_3] \quad (3.26)$$

$$= \begin{pmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{pmatrix} \quad (3.27)$$

Une matrice de rotation a les propriétés spéciales suivantes :

1. Les colonnes d'une matrice de rotation sont des vecteurs unitaires orthogonaux.
2. Les lignes d'une matrice de rotation sont également des vecteurs unitaires orthogonaux.
3. La transposée d'une matrice de rotation est son inverse, $R^T = R^{-1}$. Ainsi, $R R^T = R^T R = I$.
4. Le déterminant d'une matrice de rotation est +1.

Une rotation tridimensionnelle générale est obtenue en combinant ces matrices de rotation en une seule matrice en effectuant une multiplication matricielle. Étant donné que la multiplication matricielle n'est pas commutative, il est important de noter l'ordre des rotations telles que,

$$R_x R_y R_z = R_z R_y R_x \quad (3.28)$$

Cela implique que la rotation d'abord autour de l'axe x, puis l'axe y, puis l'axe z ne produit pas la même rotation que la rotation d'abord autour de l'axe z, puis l'axe y, puis l'axe x. Cette propriété s'accorde sur le fait que les rotations ne sont pas non plus commutatives. La matrice finale aura trois paramètres d'angle de chaque matrice de rotation, ce qui correspond à trois degrés de liberté (ces paramètres d'angle sont souvent appelés angles d'Euler $[\phi, \theta, \psi]$).

3.4. Rotations 3D à l'aide de quaternions

L'exécution d'une multiplication matrice-vecteur entre la matrice de rotation et un vecteur dans l'espace entraîne une rotation de ce vecteur dans l'espace tout en conservant sa longueur. Cela peut être exprimé comme suit :

$$\begin{aligned} \mathbf{z}' &= \mathbf{R} \mathbf{z} \\ \begin{pmatrix} z'_1 \\ z'_2 \\ z'_3 \end{pmatrix} &= \begin{pmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{pmatrix} \begin{pmatrix} z_1 \\ z_2 \\ z_3 \end{pmatrix} \end{aligned} \quad (3.29)$$

où $\mathbf{z}'$ est le vecteur $\mathbf{z}$ tourné par la matrice de rotation $\mathbf{R}$.

La possibilité de combiner plusieurs rotations en une seule matrice et la facilité d'appliquer la rotation à un vecteur font de la matrice de rotation un moyen utile et populaire de représenter les rotations même si elle est moins précise que les autres représentations.

Représentation des angles d'Euler

Les angles d'Euler sont historiquement la représentation de rotation la plus populaire où une rotation générale est décrite comme une séquence de rotations autour de trois axes de coordonnées orthogonaux, à savoir les axes x , y et z . L'idée derrière Euler Angles est de diviser une rotation en trois rotations plus simples $[\varphi, \theta, \psi]$ correspondant aux rotations sur les axes principaux et sont communément appelées tangage, lacet et roulis.

La définition des angles d'Euler n'est pas unique et les conventions dépendent du cadre de référence, des axes et de la séquence d'application des rotations. Par conséquent, pour un référentiel global, une rotation de φ dans l'axe des x suivie d'une rotation de θ dans l'axe des y fixe, puis une rotation de ψ dans l'axe des z fixe peut être représentée dans la matrice de rotation formée en pré-multipliant les rotations successives (pour le référentiel local, la post-multiplication est effectuée). Ceci peut être exprimé comme : $\mathbf{R}_{xyz} = \mathbf{R}_z(\psi)\mathbf{R}_y(\theta)\mathbf{R}_x(\varphi)$

$$= \begin{pmatrix} \cos \theta \cos \varphi & \cos \theta \sin \varphi & -\sin \theta \\ \sin \psi \cos \theta \cos \varphi & \sin \psi \cos \theta \sin \varphi & \sin \psi \sin \theta \\ -\sin \psi \sin \theta \cos \varphi & -\sin \psi \sin \theta \sin \varphi & \cos \psi \end{pmatrix} \begin{pmatrix} z_1 \\ z_2 \\ z_3 \end{pmatrix} \quad (3.30)$$

Les angles d'Euler sont devenus le moyen le plus courant de représenter la rotation d'un corps rigide dans un espace tridimensionnel car c'est le plus facile à utiliser et à comprendre. Cependant, les principaux inconvénients des angles d'Euler, tels que la sensibilité au verrouillage du cardan et la faible précision dans l'intégration des changements incrémentiels au fil du temps, ont conduit les chercheurs à utiliser d'autres représentations de rotation telles que les quaternions unitaires comme moyen de représenter efficacement les rotations.

Chapitre 3. Quaternions

Représentation Axe-Angle

Le théorème de rotation d'Euler implique que toute rotation peut être exprimée comme une rotation unique autour d'un axe. Ainsi, une expression plus généralisée d'une rotation est par la représentation axe-angle qui s'exprime simplement comme,

$$\text{rot}(\mathbf{n}, \theta), \text{ où } \mathbf{n} = n_x \mathbf{i} + n_y \mathbf{j} + n_z \mathbf{k} \quad \text{et } \theta \in [0, 2\pi] \quad (3.31)$$

L'axe de rotation $\mathbf{n}$ est défini comme tout vecteur unitaire dans l'espace tridimensionnel qui n'est pas modifié par la rotation, et l'angle θ représente la quantité de rotation autour de cet axe qui suit la règle de la main droite pour déterminer le sens positif de la rotation.

Pour faire pivoter un vecteur de $\mathbf{z}$ à $\mathbf{z}'$ en utilisant la représentation axe-angle, nous appliquons la Formule de rotation de Rodrigues définie comme,

$$\mathbf{z}' = \mathbf{z} \cos \theta + \mathbf{n}(\mathbf{n} \cdot \mathbf{z})(1 - \cos \theta) + (\mathbf{z} \times \mathbf{n}) \sin \theta \quad (3.32)$$

Notez que si l'axe $\mathbf{n}$ n'est pas l'un des axes principaux, il est généralement difficile d'identifier l'axe de rotation. De plus, combiner deux rotations successives représentées par un angle d'axe n'est pas simple et il est préférable d'utiliser une matrice de rotation ou une multiplication de quaternions pour obtenir la rotation finale, puis de la convertir en représentation d'axe.

Représentation des quaternions unitaires

Un quaternion unitaire $q = q_0 + q_1 \mathbf{i} + q_2 \mathbf{j} + q_3 \mathbf{k}$ est défini comme un quaternion avec une norme unitaire telle que,

$$\|q\| = \sqrt{q_0^2 + q_1^2 + q_2^2 + q_3^2} = 1 \quad (3.33)$$

La représentation d'une rotation par un quaternion unitaire se fait généralement en convertissant d'autres représentations de rotation en quaternions et en la normalisant. Pour une représentation axe-angle $\text{rot}(\mathbf{n}, \theta)$ où l'axe de rotation est donné par un vecteur unitaire $\mathbf{n}$ et l'angle de rotation donné par θ , sa représentation quaternion équivalente est formulée comme suit,

$$q = \cos \frac{\theta}{2} + \sin \frac{\theta}{2} (\mathbf{n} \cdot \mathbf{p}) \quad \text{et} \quad q^{-1} = \cos \frac{\theta}{2} - \sin \frac{\theta}{2} (\mathbf{n} \cdot \mathbf{p}) \quad (3.34)$$

Il est important de noter que q et $-q$ sont équivalents à la même rotation même si leurs représentations axe-angle sont différentes. Cela est dû au fait que tourner d'un angle θ autour d'un axe $\mathbf{n}$ équivaut à tourner d'un angle de $2\pi - \theta$ autour de l'axe $-\mathbf{n}$.

3.4. Rotations 3D à l'aide de quaternions

Pour les séquences de rotations qui sont représentées par plusieurs quaternions, dériver la rotation finale du quaternion est aussi simple qu'avec les matrices de rotation.

Si on nous donne une séquence de rotations de quaternions $q_1, q_2, \dots, q_n$, la rotation finale q_F est calculée en obtenant simplement le produit des quaternions en notant l'ordre de rotation et le type de référentiel.

Ainsi, pour un référentiel local, on a,

$$q_F = q_1 q_2 \dots q_n \quad (3.35)$$

Pour faire tourner un vecteur tridimensionnel p vers p' de q , le vecteur initial p est d'abord représenté comme un quaternion pur n'ayant que sa partie vectorielle comme partie imaginaire et partie réelle nulle. Afin d'obtenir p' , ce qui suit est appliqué au vecteur initial p :

$$p' = q p q^{-1}, \text{ où } q^{-1} \text{ est l'inverse de } q \quad (3.36)$$

À titre d'exemple, nous avons la rotation suivante illustrée à la Figure 3.4 où nous voulons faire pivoter le vecteur représentant le point $(2,1,1)$ le long de l'axe z , $\text{rot}(z, 90^\circ)$. Le vecteur tourné est ensuite déterminé en utilisant la formule par 90° ci-dessus et en appliquant les règles de multiplication des unités quaternion i, j et k . Cela donnera un autre quaternion pur qui représentera alors le vecteur tourné.

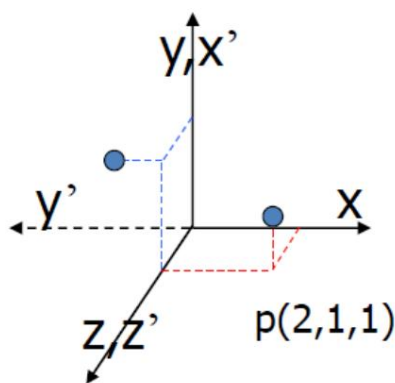


Figure 3.4 : Rotation du vecteur $(2,1,1)$ de [22]

$$p = 0 + 2i + 1j + 1k \quad (3.37)$$

$$q = \cos \frac{90^\circ}{2} + 0i + 0j + \sin \frac{90^\circ}{2} k = \frac{\sqrt{2}}{2} + \frac{\sqrt{2}}{2} k \quad (3.38)$$

$$q^{-1} = \cos \frac{90^\circ}{2} + 0i + 0j - \sin \frac{90^\circ}{2} k = \frac{\sqrt{2}}{2} - \frac{\sqrt{2}}{2} k \quad (3.39)$$

$$p' = q p q^{-1} = -i + 2j + k \quad (3.40)$$

Par conséquent, les coordonnées du vecteur $(2,1,1)$ tourné de 90° le long de l'axe z est $(-1,2,1)$.

Chapitre 3. Quaternions

3.4.2 Conversions de quaternions

En traitant des quaternions unitaires, il est nécessaire d'apprendre à convertir d'autres représentations de rotation en quaternions unitaires et inversement. Dans cette étude, nous limitons nos conversions uniquement à ce que nous utiliserons pour dériver la cinématique du robot en appliquant le concept de quaternions⁶.

Quaternion unitaire = Matrice de rotation

La conversion d'un quaternion $q = q_0 + q_1i + q_2j + q_3k$ en une matrice de rotation R est donné par la matrice suivante,

$$R(q) := \begin{pmatrix} q_0^2 + q_1^2 - q_2^2 - q_3^2 & 2q_1q_2 + 2q_0q_3 & 2q_1q_3 - 2q_0q_2 & 2q_2q_3 \\ 2q_1q_2 - 2q_0q_3 & q_0^2 - q_1^2 + q_2^2 - q_3^2 & 2q_0q_1 + 2q_2q_3 & 2q_0q_2 + 2q_1q_3 \\ 2q_1q_3 + 2q_0q_2 & 2q_0q_1 - 2q_2q_3 & q_0^2 - q_1^2 - q_2^2 + q_3^2 & 2q_1q_2 - 2q_0q_3 \\ 2q_2q_3 - 2q_0q_1 & 2q_0q_2 + 2q_1q_3 & 2q_0q_3 - 2q_1q_2 & q_0^2 + q_1^2 + q_2^2 - q_3^2 \end{pmatrix} \quad (3.41)$$

Chaque élément de la matrice R est noté r_{ij} pour $i, j = 1, 2, 3$.

Quaternion unitaire = Matrice de rotation

Pour obtenir la cartographie inverse, en inspectant la relation des éléments de R avec les coefficients du quaternion (q_0, q_1, q_2, q_3) conduira à quatre cartographies inverses différentes. Certaines de ces cartographies produiront des résultats complexes.

Par conséquent, nous utilisons les conditions suivantes (selon les paramètres de R) pour déterminer quel mappage utiliser :

a) Si $r_{22} > -r_{33}$, $r_{11} > -r_{22}$, $r_{11} > -r_{33}$, alors,

$$q(R) = \frac{1}{2} \begin{pmatrix} (1 + r_{11} + r_{22} + r_{33})^{\frac{1}{2}} \\ (r_{23} - r_{32}) / (1 + r_{11} + r_{22} + r_{33})^{\frac{1}{2}} \\ (r_{31} - r_{13}) / (1 + r_{11} + r_{22} + r_{33})^{\frac{1}{2}} \\ (r_{12} - r_{21}) / (1 + r_{11} + r_{22} + r_{33})^{\frac{1}{2}} \end{pmatrix} \quad (3.42)$$

b) Si $r_{22} < -r_{33}$, $r_{11} > r_{22}$, $r_{11} > r_{33}$, alors,

$$q(R) = \frac{1}{2} \begin{pmatrix} (r_{23} - r_{32}) / (1 + r_{11} - r_{22} - r_{33})^{\frac{1}{2}} \\ (1 + r_{11} - r_{22} - r_{33})^{\frac{1}{2}} \\ (r_{12} + r_{21}) / (1 + r_{11} - r_{22} - r_{33})^{\frac{1}{2}} \\ (r_{31} + r_{13}) / (1 + r_{11} - r_{22} - r_{33})^{\frac{1}{2}} \end{pmatrix} \quad (3.43)$$

⁶Dans notre application, nous avons particulièrement utilisé les conversions de la matrice de rotation et de la représentation axe-angle en quaternions unitaires.

3.4. Rotations 3D à l'aide de quaternions

c) Si $r_{22} > r_{33}$, $r_{11} < r_{22}$, $r_{11} < -r_{33}$, alors,

$$q(R) = \frac{1}{2} \begin{pmatrix} (r_{31} - r_{13})/(1 - r_{11} + r_{22} - r_{33}) & \frac{1}{2} \\ (r_{12} + r_{21})/(1 - r_{11} + r_{22} - r_{33}) & \frac{1}{2} \\ r_{11} + r_{22} - r_{33} & \frac{1}{2} \\ (r_{23} + r_{32})/(1 - r_{11} + r_{22} - r_{33}) & \frac{1}{2} \end{pmatrix} \quad (3.44)$$

d) Si $r_{22} < r_{33}$, $r_{11} < -r_{22}$, $r_{11} < r_{33}$, alors,

$$q(R) = \frac{1}{2} \begin{pmatrix} (r_{12} - r_{21})/(1 - r_{11} - r_{22} + r_{33}) & \frac{1}{2} \\ (r_{31} + r_{13})/(1 - r_{11} - r_{22} + r_{33}) & \frac{1}{2} \\ (r_{23} + r_{32})/(1 - r_{11} - r_{22} + r_{33}) & \frac{1}{2} \\ r_{11} - r_{22} + r_{33} & \frac{1}{2} \end{pmatrix} \quad (3.45)$$

Choisir parmi ces mappages permet de réduire l'imprécision en évitant les situations où le dénominateur est proche de zéro.

Unité Quaternion = Axe-Angle

Le quaternion q équivalent d'une rotation θ autour d'un vecteur unitaire n dans un espace tridimensionnel représenté par l'angle d'axe $\text{rot}(n, \theta)$ est défini comme,

$$q(\theta, n) := \left(\cos\left(\frac{\theta}{2}\right), n \sin\left(\frac{\theta}{2}\right) \right) \quad (3.46)$$

Notez que nous ne considérons que les vecteurs unitaires, $\|n\| = 1$, dans cette conversion.

Unité Quaternion = Axe-Angle

Pour l'application inverse du quaternion unitaire q à la représentation axe-angle $\text{rot}(n, \theta)$, nous considérons d'abord $q = q_1 q_2 q_3$ qui est la partie vectorielle de q . La représentation axe-angle peut alors être calculée par :

$$\theta = 2 \arccos(q_0) \quad (3.47)$$

$$n = \frac{q}{\|q\|} \quad (3.48)$$

Ainsi, l'angle de rotation θ peut être facilement calculé à partir du premier terme du quaternion unitaire donné tandis que l'axe de rotation n peut être dérivé en normalisant simplement la partie vectorielle (partie imaginaire) du quaternion unitaire.

3.5 Quaternions doubles

Les quaternions sont capables de représenter des rotations tridimensionnelles comme nous l'avons vu dans la discussion précédente. Cependant, lorsqu'il s'agit de transformations en trois dimensions, il pourrait également y avoir un déplacement ou une translation dans laquelle notre forme actuelle de quaternion n'est pas en mesure de représenter. Pour cette section, nous introduirons le concept de Quaternions Doubles qui dépasse cette limite et sera utilisé pendant toute la durée de l'étude.

3.5.1 Définition

Les doubles quaternions permettent de représenter à la fois la rotation et la translation dans un seul vecteur de transformation. Pour un quaternion unitaire q qui représente la rotation appropriée et pour un vecteur p écrit comme un quaternion pur qui représente le déplacement correspondant, la forme la plus couramment utilisée en robotique d'un quaternion dual est donnée par la paire quaternion-vecteur,⁷

$$Q(q, p) = \cos \frac{\theta}{2} + \frac{\theta}{2} \text{, péché} \frac{\theta}{2} \langle kx, ky, kz \rangle + \langle px, py, pz \rangle \quad (3.49)$$

Pour la partie quaternion, nous utilisons la conversion d'une représentation de rotation axe-angle en un quaternion unitaire. Ainsi, θ donne l'angle de rotation et $\langle kx, ky, kz \rangle$ désigne l'axe de rotation. La partie vectorielle est simplement le vecteur de déplacement donné par $p = \langle px, py, pz \rangle$.

Clairement, pour une transformation sans rotation et uniquement avec déplacement, le quaternion dual est représenté par :

$$Q(q, p) = ([1, \langle 0, 0, 0 \rangle], \langle px, py, pz \rangle) \quad (3.50)$$

où $[1, \langle 0, 0, 0 \rangle]$ représente le quaternion d'identité unitaire.

3.5.2 Inverse d'un double quaternion

L'inverse d'un quaternion dual $Q(q, p)$ est noté :

$$Q^{-1} = ([q^{-1}], \langle -q^{-1} p \rangle) \quad (3.51)$$

quaternion⁻¹ désigne l'inverse habituel du quaternion q . Par conséquent, pour a où $q = [s, v]$, la partie vectorielle peut également s'écrire,

$$q^{-1} p = -p + [-2s(v \times (-p)) + 2v \times (v \times (-p))] \quad (3.52)$$

Notez que p est un quaternion pur (partie réelle nulle) et est donc également un vecteur.

⁷Les quaternions duaux sont généralement définis à l'aide de nombres duaux tels que $Q = q + 2$ où $p = 0$. Ceci est clairement décrit dans [14].

3.5.3 Multiplication d'un double quaternion

Dans les quaternions doubles, la séquence de rotations et de déplacements est représentée en multipliant leurs représentations de quaternions doubles correspondantes. Ainsi, pour les transformations $Q1(q1, p1)$ suivies de $Q2(q2, p2)$ pour un repère de coordonnées local, leur produit dual quaternion est donné par :

$$Q1Q2 = (q1, p1) (q2, p2) = ([q1 \ q2], \langle q1 \ p2 \ q1^{-1} + p1 \rangle) \quad (3.53)$$

où la partie quaternion est simplement la multiplication entre les quaternions $q1$ et $q2$ et la partie vecteur peut également être écrite comme,

$$q1 \ p2 \ q1^{-1} + p1 = p2 + 2s1(v1 \times p2) + 2v1 \times (v1 \times p2) + p1 \quad (3.54)$$

où $q1 = [s1, v1]$ et $p2$ est un quaternion pur et agit également comme un vecteur.

L'analyse de l'opération pour la partie traduction du quaternion double montre que la traduction suivante est d'abord tournée par la rotation précédente et la traduction tournée résultante est ensuite ajoutée à la traduction actuelle.

3.5.4 Transformation par un double quaternion

La transformation d'un vecteur à l'aide de quaternions doubles suit presque le même processus que la multiplication de deux quaternions doubles. Si nous avons la transformation finale donnée par le quaternion dual,

$$QF = ([w, \langle a, b, c \rangle], \langle px, py, pz \rangle) \quad (3.55)$$

où $[w, \langle a, b, c \rangle] = qF$ et $\langle px, py, pz \rangle = pF$, et un vecteur initial à transformer donné par,

$$v = \langle vx, vy, vz \rangle \quad (3.56)$$

Nous obtenons le vecteur transformé (tourné et translaté) v' par, $v' = \langle v'_x, v'_y, v'_z \rangle$

$$v' = (qF \ v \ qF^{-1}) + pF \quad (3.57)$$

Par conséquent, un vecteur est transformé à l'aide de quaternions doubles en le faisant d'abord tourner à l'aide de la partie quaternion qF (représentant la rotation) du double quaternion, puis en le traduisant en y ajoutant la partie vectorielle pF (représentant la translation) du quaternion double. Ce processus est similaire à la façon dont une transformation est appliquée à l'aide de matrices de transformation.

Dans le chapitre suivant, nous démontrerons l'application des quaternions doubles en les utilisant dans la dérivation de la cinématique avant et inverse d'un robot avec une application pratique sur le robot universel UR3.

Chapitre 4

Application des quaternions à Robot Universel UR3

L'objectif principal de cette étude est de pouvoir appliquer le concept de quaternions dans la manipulation de mouvements de robots. Pour cette étude, nous travaillerons avec le robot universel UR3 et dériverons sa cinématique directe et inverse à l'aide de quaternions. A la fin de ce chapitre, une application réelle de programmation du calcul pour la cinématique de UR3 est testée et présentée.

4.1 Robot Universel - UR3

L'Universal Robot UR3 est un bras robotique collaboratif. Il a six degrés de liberté (6DOF) et est composé de six articulations rotoïdes, communément appelées (1) base, (2) épaule, (3) coude, (4) poignet 1, (5) poignet 2 et (6) poignet 3, chacun avec une plage de travail de 360 degrés.



Figure 4.1 : Robot universel UR3

¹Les spécifications techniques du robot se trouvent à l'annexe A.

4.2 Cinématique directe par double quaternions

La cinématique directe est généralement plus facile à calculer que son homologue et il en va de même pour son calcul par Dual Quaternions. Notre objectif est de déterminer l'orientation et la position finales de l'effecteur compte tenu de tous les angles articulaires. Fondamentalement, il existe au moins deux façons de dériver la cinématique directe à l'aide de doubles quaternions. Elle peut être soit à partir d'une matrice de rotation, soit à partir d'une représentation axe-angle. Chacun de ces deux résultats entraînera la même rotation finale, mais chacun utilise des coordonnées différentes pour la position TCP initiale. Dans cette étude, nous utiliserons l'approche avec la représentation axe-angle.²

4.2.1 Représentations de la transformation des joints

Compte tenu de la structure cinématique du robot UR3 en position zéro, nous dérivons les quaternions doubles représentant la transformation dans chaque articulation en dérivant d'abord la représentation axe-angle de chaque articulation par rapport au cadre de base et en la définissant comme la première partie de notre double quaternion et la deuxième partie en obtenant la translation du cadre commun courant vers le suivant³ :

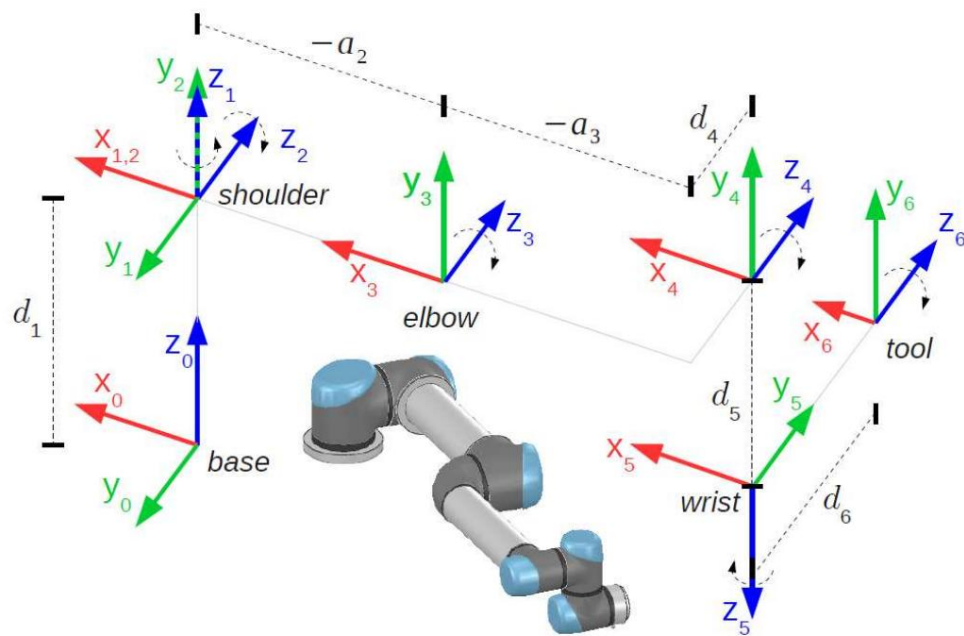


Figure 4.2 : Structure cinématique de UR3 en position zéro ($\theta_{1,2,3,4,5,6} = 0$) [3].

²Cette étude suit l'approche faite dans [4] et [15]

³La convention habituelle de dérivation de la cinématique pour UR3 peut être trouvée dans [3] et [11]

4.2. Cinématique directe par doubles quaternions

Notez que l'axe z représente l'axe de rotation et nous suivons le règle de la main droite pour déterminer le sens positif de la rotation sur chaque axe de rotation. Nous allons représenter les axes z par rapport à la coordonnée l'axe du cadre de base.

Pour l'UR3, nous avons également les paramètres DH suivants :4

j^e	α_{i-1}	a_{i-1}	d_i	θ_i
1	0	0	d_1	θ_1
2	$\alpha_1 = 90^\circ$	0	0	θ_2
3	0	a_2		
4	0	a_3	d_4	θ_4
5	$\alpha_4 = 90^\circ$	0	d_5	θ_5
6	$\alpha_5 = -90^\circ$	0	d_6	θ_6

Tableau 4.1 : Paramètres UR3 Denavit-Hartenberg

Pour la première articulation, dans la partie rotation, nous avons l'angle de rotation sous la forme θ_1 et nous avons notre axe de rotation z_1 aligné avec l'axe de base z_0 . Pour le partie de translation, partant du cadre de base vers le cadre d'épaule, il y a est un déplacement de d_1 dans l'axe z. Par conséquent, notre double quaternion pour le première articulation est représentée par :

$$Q_1 = \cos \frac{\theta_1}{2} \text{ , péché } \frac{\theta_1}{2} \langle 0, 0, 1 \rangle, \langle 0, 0, d_1 \rangle$$

Pour la deuxième articulation, l'angle de rotation est θ_2 et notre axe de rotation z_2 est maintenant parallèle à l'axe $-y_0$ du repère de base. Pour la partie traduction, nous devons analyser que la rotation de la deuxième articulation entraînera une translation en axe x_0 et en axe z_0 qui dépend de a_2 et θ_2 . Donc, nous avons:

$$Q_2 = \cos \frac{\theta_2}{2} \text{ , péché } \frac{\theta_2}{2} \langle 0, -1, 0 \rangle, \langle a_2 \cos \theta_2, 0, a_2 \sin \theta_2 \rangle$$

Pour la troisième articulation, l'angle de rotation est θ_3 et notre axe de rotation z_3 , identique à z_2 , est parallèle à l'axe $-y_0$ du repère de base. Pour la traduction partie, la rotation de la troisième articulation entraînera une translation dans l'axe x_0 et l'axe z_0 qui dépend de a_3 et θ_3 . En dehors de cela, pour se rendre au quatrième articulation, il y a aussi un déplacement de d_4 dans l'axe $-y_0$. Ainsi:

$$Q_3 = \cos \frac{\theta_3}{2} \text{ , péché } \frac{\theta_3}{2} \langle 0, -1, 0 \rangle, \langle a_3 \cos \theta_3, -d_4, a_3 \sin \theta_3 \rangle$$

Les paramètres 4DH sont discutés et illustrés dans la section 4.4.1

Chapitre 4. Application des quaternions à Universal Robot UR3

Pour la quatrième articulation, l'angle de rotation est θ_4 et notre axe de rotation z_4 , comme z_2 et z_3 , est parallèle à l'axe $-y_0$ du repère de base. Pour la partie translation, la rotation de la quatrième articulation entraînera une translation dans l'axe x_0 et l'axe z_0 qui dépend de d_5 et θ_5 . Par conséquent, nous avons :

$$Q_4 = \cos \frac{\theta_4}{2} + \sin \frac{\theta_4}{2} \langle 0, -1, 0 \rangle, \langle d_5 \sin \theta_4, 0, -d_5 \cos \theta_4 \rangle$$

Pour la cinquième articulation, l'angle de rotation est θ_5 et notre axe de rotation z_5 est parallèle à l'axe $-z_0$ du repère de base. Pour la partie translation, la rotation de la cinquième articulation entraînera une translation dans l'axe x_0 et l'axe y_0 qui dépend de d_6 et θ_5 . Par conséquent, nous avons :

$$Q_5 = \cos \frac{\theta_5}{2} + \sin \frac{\theta_5}{2} \langle 0, 0, -1 \rangle, \langle -d_6 \sin \theta_5, -d_6 \cos \theta_5, 0 \rangle$$

Pour la sixième et dernière articulation, l'angle de rotation est θ_6 et notre axe de rotation z_6 est parallèle à l'axe $-y_0$ du repère de base. Il n'y a pas de traduction puisqu'il s'agit de la dernière image. Par conséquent, nous avons :

$$Q_6 = \cos \frac{\theta_6}{2} + \sin \frac{\theta_6}{2} \langle 0, -1, 0 \rangle, \langle 0, 0, 0 \rangle$$

La représentation des transformations par des quaternions doubles a un concept similaire avec les matrices de transformation. La principale différence est que les quaternions doubles sont plus compacts de sorte que nous n'avons besoin que de 7 composants alors que pour les matrices de transformation, nous utilisons $4 \times 4 = 16$ composants.

4.2.2 Calcul de la cinématique directe UR3

Après avoir dérivé les quaternions doubles correspondants pour chaque articulation, nous prenons leurs produits du cadre de base au cadre final pour obtenir le quaternion double final représentant la transformation finale et la cinématique avant du robot. Ainsi, nous avons,

$$Q_{FK} = Q_1 Q_2 Q_3 Q_4 Q_5 Q_6 \quad (4.1)$$

Pour cette approche, les coordonnées du point à transformer doivent également être par rapport au repère de base. Ainsi, lors de la transformation, le point est tourné de la partie rotation de Q_{FK} puis translaté de sa partie translation à partir du repère de base.

4.3 Cinématique inverse par doubles quaternions

Dériver la cinématique inverse nécessite plus de travail et d'analyse que la cinématique directe. Nous devons appliquer le concept de transformations par des quaternions doubles, traiter des équations trigonométriques complexes et analyser géométriquement les transformations effectuées par chaque articulation sur le robot et comment chacune d'elles affecte les coordonnées des cadres articulaires suivants. Notre objectif en cinématique inverse est de déterminer les angles articulaires nécessaires pour déplacer l'effecteur terminal vers la position et l'orientation souhaitées qui sont spécifiées par la transformation finale, $QFK = ([w, a, b, c], [px, py, pz])$.

4.3.1 Représentations des transformations articulaires

Pour commencer par la dérivation de la cinématique inverse, nous avons besoin des quaternaires duaux représentant la transformation effectuée par chaque articulation Q_i , pour $1 \leq i \leq 6$, que nous avons déjà dérivés pour la cinématique directe ci-dessus, ainsi que de leurs inverses quaternioniques duaux correspondants, Q_i^{-1} .

À l'avenir, nous utiliserons les représentations suivantes pour les équations longues : $\cos \theta_i \rightarrow c_i$, $\sin \theta_i \rightarrow s_i$, parce que $\frac{\theta_i}{2} \rightarrow c_i^{-1}$, $\sin \theta_i \rightarrow s_i^{-1}$.

Ainsi, pour chaque articulation, nous avons :

$$\begin{aligned} Q_1 &= ([c_1 + s_1k], [d_1k]) \\ Q_2 &= ([c_2 - s_2j], [a_2c_2i + a_2s_2k]) \\ Q_3 &= ([c_3 - s_3j], [a_3c_3i - d_4j + a_3s_3k]) \\ Q_4 &= ([c_4 - s_4j], [d_5s_4i - d_5c_4k]) \\ Q_5 &= ([c_5 - s_5k], [-d_6s_5i - d_6c_5j]) \\ Q_6 &= ([c_6 - s_6j], [0]) \end{aligned}$$

et les inverses comme :

$$\begin{aligned} Q_1^{-1} &= ([c_1 - s_1k], [-d_1k]) = \\ Q_2^{-1} &= ([c_2 + s_2j], [-a_2c_2i - a_2s_2k]) \\ Q_3^{-1} &= ([c_3 + s_3j], [-a_3c_3i + d_4j - a_3s_3k]) = ([c_4 \\ Q_4^{-1} &= ([c_4 + s_4j], [d_5s_4i + d_5c_4k]) \\ Q_5^{-1} &= ([c_5 + s_5k], [d_6s_5i + d_6c_5j]) = \\ Q_6^{-1} &= ([c_6 + s_6j], [0]) \end{aligned}$$

Nous utiliserons les représentations duelles quaternions de ces deux articulations pour obtenir la relation entre les combinaisons de transformations successives.

4.3.2 Produits de transformation des joints

En utilisant les quaternions doubles Q_i ($1 \leq i \leq 6$), qui dénote la cinématique transformations qui décrivent les relations spatiales entre les cadres de coordonnées et Q_i^{-1} comme leurs inverses correspondants, nous définissons d'abord le produit double quaternion M_i comme :

$$M_i = Q_i \quad M_{i+1} \text{ où } 1 \leq i \leq 6 \quad (4.2)$$

Pour $i = 6$, on a que $M_6 = Q_6$. Par conséquent, nous avons M_i comme,

$$\begin{aligned} M_6 &= Q_6 \\ M_5 &= Q_5 \quad M_6 = Q_5 \quad Q_6 \\ M_4 &= Q_4 \quad M_5 = Q_4 \quad Q_5 \quad Q_6 \\ M_3 &= Q_3 \quad M_4 = Q_3 \quad Q_4 \quad Q_5 \quad Q_6 \\ M_2 &= Q_2 \quad M_3 = Q_2 \quad Q_3 \quad Q_4 \quad Q_5 \quad Q_6 \\ M_1 &= Q_1 \quad M_2 = Q_1 \quad Q_2 \quad Q_3 \quad Q_4 \quad Q_5 \quad Q_6 \end{aligned}$$

Ensuite, nous définissons un autre produit de quaternion dual N_i comme suit :

$$N_{i+1} = Q_i^{-1} \quad N_i \text{ où } 1 \leq i \leq 6 \quad (4.3)$$

Pour $i = 0$, nous définissons N_1 comme le quaternion de la transformation finale, ce qui signifie,

$$N_1 = [Rw, Tw] = ([w, < a, b, c >], < px, py, pz >)$$

Par conséquent, nous avons N_i comme,

$$\begin{aligned} N_1 &= [Rw, Tw] \\ N_2 &= Q_1^{-1} \quad N_1 = Q_1^{-1} \quad [Rw, Tw] \\ N_3 &= Q_2^{-1} \quad N_2 = Q_2^{-1} \quad Q_1^{-1} \quad [Rw, Tw] \\ N_4 &= Q_3^{-1} \quad N_3 = Q_3^{-1} \quad Q_2^{-1} \quad Q_1^{-1} \quad [Rw, Tw] \\ N_5 &= Q_4^{-1} \quad N_4 = Q_4^{-1} \quad Q_3^{-1} \quad Q_2^{-1} \quad Q_1^{-1} \quad [Rw, Tw] \\ N_6 &= Q_5^{-1} \quad N_5 = Q_5^{-1} \quad Q_4^{-1} \quad Q_3^{-1} \quad Q_2^{-1} \quad Q_1^{-1} \quad [Rw, Tw] \end{aligned}$$

Clairement, nous pouvons voir que M_1 et N_1 sont juste équivalents à la cinématique avant QFK. Une analyse plus poussée de la composition de chacune de ces transformations entre les produits de quaternions duaux M et N implique que $M_i = N_i$. Donc, leurs termes correspondants doivent également être égaux.

L'étape suivante consiste à assimiler chaque terme de quaternion double correspondant entre M_i et N_i qui nous donneront plusieurs équations que nous pourrions utiliser pour calculer pour la cinématique inverse.

4.3.3 Transformations des cadres de la base aux poignets

Calculer pour les produits de transformation M_i et N_i où $(a \leq i \leq b)$ nous conduira à $((b - a) + 1) \times 7 \times 2$ équations où les multiplicateurs représentent le nombre de transformations $(b - (a - 1))$, le nombre de termes dans le quaternion dual (7), et pour M et N en conséquence. Nous égalisons les termes pour chaque produit de transformation correspondant et obtenons $((b - a) + 1) \times 7$ équations et nous essayons de trouver une relation entre certaines de ces équations et de les manipuler pour pouvoir résoudre explicitement les angles articulaires.

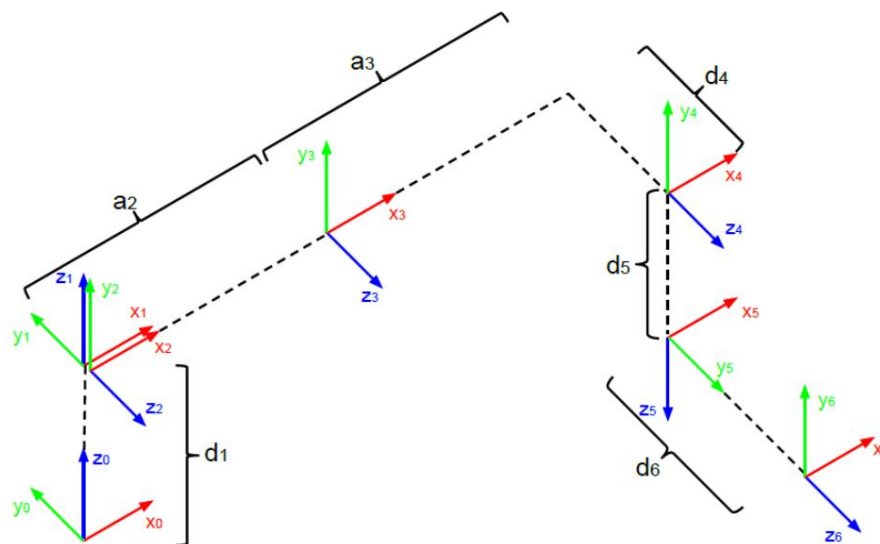


Figure 4.3 : Cadres de coordonnées de UR3 ($\theta_1, 2, 3, 4, 5, 6 = 0$) de [11].

Transformation de l'image 1 (base) à l'image 6 (poignet 3)

Suivant [4] et [15], nous devrions commencer par la transformation de la base au poignet 3. Cela nous mènera à 84 équations où $(1 \leq i \leq 6)$. Nous assimilons les produits et termes de transformation correspondants entre M_i et N_i .

Ces équations sont répertoriées dans l'annexe B. La facilité d'obtenir une équation simple à calculer pour les angles d'articulation dépend de la structure du robot.

Dans le cas de UR3, il n'y a pas d'équations qui peuvent directement dériver la valeur de n'importe quel angle d'articulation. Nous pouvons trouver une équation pour θ_5 puis θ_6 mais nous avons besoin de θ_1 pour les résoudre. Nous pouvons appliquer des méthodes numériques pour estimer les valeurs des angles articulaires mais dans notre cas, nous avons besoin de valeurs exactes. Pour résoudre ce problème, nous pouvons ensuite incorporer l'approche géométrique qui est couramment utilisée pour dériver la cinématique inverse d'un robot et passer à la transformation suivante qui est la transformation de la base au poignet 2.

Chapitre 4. Application des quaternions à Universal Robot UR3

Transformation de l'image 1 (base) à l'image 5 (poignet 2)

Notre principale motivation pour passer à cette transformation est de résoudre pour θ_1 . L'approche géométrique consiste à trouver la relation entre les angles des articulations et les déplacements. La figure ci-dessous montre la relation entre l'angle d'articulation θ_1 avec les coordonnées du cadre 1 (base) et du cadre 5 (poignet 2).

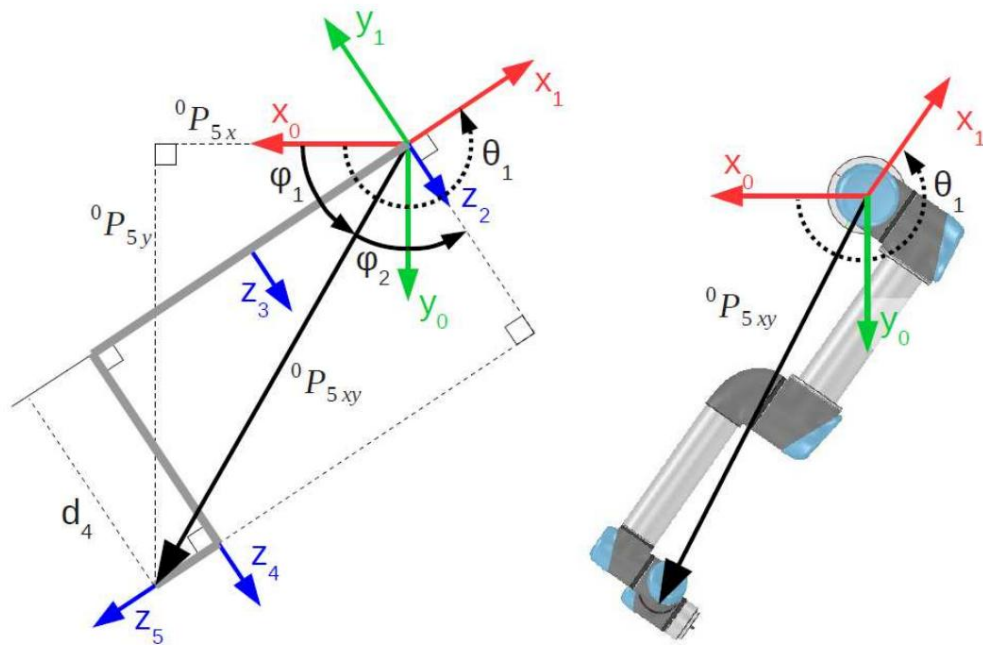


Figure 4.4 : Structure du robot UR3 du cadre 1 au cadre 5 (de [3])

Comme on peut le voir, il sera possible de déterminer la valeur de θ_1 en utilisant le déplacement entre la Frame 1, Q1, et la Frame 5, Q5. La transformation de la base au poignet 2 nous mènera à 70 équations où $(1 \leq i \leq 5)$. Heureusement, au lieu de recalculer ces transformations, on peut utiliser la transformation précédente et prendre N1 à N5 puisqu'elles sont juste équivalentes. La seule différence est que nous travaillons sur les coordonnées à l'image 5. Ainsi, nous n'aurons qu'à faire une traduction des coordonnées de l'image 6 à l'image 5. En ce qui concerne les produits de transformation vers l'avant, nous devrions commencer par M5 au lieu de M6. Cela peut être fait simplement en réglant $M5 = Q5$ et la traduction à zéro. L'équivalence des termes de M5 avec ses termes correspondants dans N5 et l'application d'identités trigonométriques nous donneront notre formule pour θ_1 .

Cela nous permettra ensuite de résoudre pour θ_5 et θ_6 en utilisant les équations que nous avons obtenues de la transformation du cadre 1 (base) au cadre 6 (poignet 3).

4.3. Cinématique inverse par doubles quaternions

Dérivation de l'angle d'articulation θ_1 (base)

Pour dériver une formule pour θ_1 , nous pouvons traduire les coordonnées d'abord dans l'image finale (poignet 3) vers l'image 5 (poignet 2) et les utiliser dans la formule à extraire des produits de transformation. Pour revenir à l'image 5, nous analysons les transformations par le poignet 3 (Q6) et le poignet 2 (Q5) et leurs contributions à la transformation finale QFK.

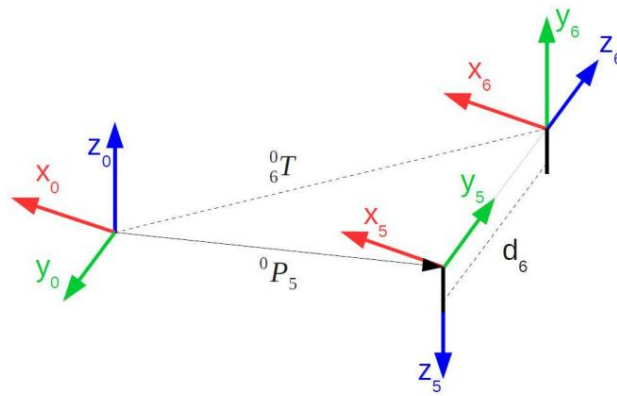


Figure 4.5 : Translation du cadre 6 (poignet 3) au cadre 5 (poignet 2) [3]

Transformation de l'image 5 (poignet 2) :

$$Q5 = ([\bar{c}5 - s\bar{5}k], < -d6s5i - d6c5j >)$$

Transformation de l'image 6 (poignet 3) :

$$Q6 = ([\bar{c}6 - s\bar{6}j], < 0 >)$$

Transformation finale : Image 1 (Base) à Image 6 (Poignet 3)

$$QFK = Q1 \quad Q2 \quad Q3 \quad Q4 \quad Q5 \quad Q6 = Q1 \quad 6$$

De toute évidence, il n'y a pas de traductions effectuées par Q6, seulement une rotation. Par conséquent, notre position d'effecteur final est déjà déterminée dans la transformation jusqu'à Q5. Ainsi, pour la partie translation (p) de QFK, nous avons,

$$\begin{aligned} (QFK)p &= (Q1 \quad 6)p = (Q1 \quad 5)p \\ &= (Q1 \quad Q2 \quad Q3 \quad Q4 \quad Q5)p \end{aligned} \quad (4.4)$$

Notez que ce dont nous avons besoin, ce sont les coordonnées à l'image 5 et nous ne traitons que de la partie translation du quaternion double.

Chapitre 4. Application des quaternions à Universal Robot UR3

Nous devons annuler la traduction effectuée par Q5. Nous le faisons en utilisant son inverse,

$$Q5^{-1} = ([-c5 + s5k], < d6i >)$$

Donc,

$$\begin{aligned} (QFK \ Q^{-1}5) &= (Q1 \ Q2 \ Q3 \ Q4 \ Q5 \ Q^{-1}5)p \\ &= (Q1 \ Q2 \ Q3 \ Q4)p \\ &= (Q14)p \end{aligned} \quad (4.5)$$

Nous n'avons pas encore de valeur pour θ_5 donc nous ne pouvons pas complètement construire Q5, mais puisque nous ne traitons que de la partie traduction, nous n'en avons pas vraiment besoin. Au lieu de cela, nous pouvons définir une valeur arbitraire pour sa partie quaternion et définir uniquement la partie traduction. Rappelons que la détermination de la partie translation à partir d'une double multiplication de quaternions est définie sans utiliser qB,

$$(QA \ QB)p = < qA \ pB \ q^{-1}_{UN} + pA > \quad (4.6)$$

Puisque $(QFK)p = (Q15)p$, on peut avoir,

$$(QFK \ Q^{-1}5)p = qFK \ p = < x5i + y5j \ q^{-1}_{FK} + pFK > \quad (4.7)$$

$$+ z5k >= (Q14)p \quad (4.8)$$

où $< x5, y5, z5 >$ sont les coordonnées dans l'image 5 par rapport à l'image 1.

Maintenant, pour calculer pour θ_1 , nous avons besoin des produits de transformation de quaternions doubles pour la transformation de la base au poignet 2. Comme nous l'avons indiqué dans notre analyse, nous pouvons simplement utiliser (B.13) $M56 = N56$, où le terme $\cos \theta_5$ est exclu.

Donc,

$$\begin{aligned} 0 &= y \cos \theta_1 - x \sin \theta_1 + d4 \ y \\ \cos \theta_1 - x \sin \theta_1 &= -d4 \end{aligned} \quad (4.9)$$

On applique l'identité trigonométrique suivante,

$$a \sin \theta + b \cos \theta = c$$

$$\theta = \text{atan2}(a, b) \pm \text{atan2}(\sqrt{a^2 + b^2 - c^2}, c) \quad (4.10)$$

Et maintenant, nous utilisons les coordonnées du cadre 5 pour dériver la formule de θ_1 ,

$$\theta_1 = \text{atan2}(-x5, y5) \pm \text{atan2}(\sqrt{x5^2 + y5^2 - d4^2}, -d4) \quad (4.11)$$

Nous considérons deux valeurs pour θ_1 correspondant à la rotation de l'articulation de base consistant à régler l'épaule vers la gauche ou vers la droite.

4.3. Cinématique inverse par doubles quaternions

Dérivation de l'angle d'articulation θ_5 (poignet 2)

Maintenant qu'on nous donne θ_1 , nous pouvons utiliser la même paire d'équations du dual produits de transformation quaternion pour la transformation de la base au poignet 3

(B.13) mais sans exclure aucun terme pour dériver l'équation de θ_5 ,

$$\begin{aligned} \underline{M56} &= \underline{N56} \\ -d_6 \cos \theta_5 &= y \cos \theta_1 - x \sin \theta_1 + d_4 \end{aligned} \quad (4.12)$$

Nous pouvons calculer directement pour θ_5 par,

$$\theta_5 = \pm \arccos \frac{x_6 \sin \theta_1 - y_6 \cos \theta_1 - d_4}{d_6} \quad (4.13)$$

Nous considérons deux valeurs de θ_5 pour la rotation articulaire du Poignet 2 correspondant au poignet soit vers le haut ou vers le bas.

Dérivation de l'angle d'articulation θ_6 (poignet 3)

On a maintenant θ_1 et θ_5 , on peut alors utiliser les couples d'équations (B.29), (B.30), (B.31) et (B.32) également à partir des produits de transformation de quaternions duaux pour la transformation de la base au poignet 3 pour dériver l'équation de θ_6 ,

$$\begin{aligned} \underline{M21} &= \underline{N21} \\ \text{parce que } \frac{\theta_5}{2} \quad \text{parce que } \frac{\theta_2 + \theta_3 + \theta_4 + \theta_6}{2} &= w \cos \frac{\theta_1}{2} + c \sin \frac{\theta_1}{2} \end{aligned} \quad (4.14)$$

$$\begin{aligned} \underline{M22} &= \underline{N22} \\ \sin \frac{\theta_5}{2} \quad \sin \frac{\theta_2 + \theta_3 + \theta_4 - \theta_6}{2} &= w \sin \frac{\theta_1}{2} + b \sin \frac{\theta_1}{2} \end{aligned} \quad (4.15)$$

$$\begin{aligned} \underline{M23} &= \underline{N23} \\ -\cos \frac{\theta_5}{2} \quad \sin \frac{\theta_2 + \theta_3 + \theta_4 + \theta_6}{2} &= -w \sin \frac{\theta_1}{2} + b \cos \frac{\theta_1}{2} \end{aligned} \quad (4.16)$$

$$\begin{aligned} \underline{M24} &= \underline{N24} \\ -\sin \frac{\theta_5}{2} \quad \sin \frac{\theta_2 + \theta_3 + \theta_4 - \theta_6}{2} &= -w \sin \frac{\theta_1}{2} + c \cos \frac{\theta_1}{2} \end{aligned} \quad (4.17)$$

Chapitre 4. Application des quaternions à Universal Robot UR3

En utilisant des identités trigonométriques (somme des angles), celles-ci sont équivalentes à,

$$c^-(2+3+4)c^-6 - s^-(2+3+4)s^-6 = \frac{wc^-1 + cs^-1}{c^-5} = \alpha \quad (4.18)$$

$$s^-(2+3+4)c^-6 - c^-(2+3+4)s^-6 = \frac{ac^-1 + bs^-1}{s^-5} = \beta \quad (4.19)$$

$$s^-(2+3+4)c^-6 + c^-(2+3+4)s^-6 = \frac{\text{comme}^-1 - bc^-1}{c^-5} = \gamma \quad (4.20)$$

$$c^-(2+3+4)c^-6 + s^-(2+3+4)s^-6 = \frac{ws^-1 - cc^-1}{s^-5} = \delta \quad (4.21)$$

Multiplier (4.18) par s^-6 nous donnera,

$$c^-(2+3+4)c^-6s^-6 - s^-(2+3+4)s^-6 \frac{2}{6} = \alpha s^-6 \quad (4.22)$$

Et multiplier (4.19) par c^-6 nous donnera :

$$s^-(2+3+4)c^-6 \frac{2}{6} - c^-(2+3+4)s^-6c^-6 = \beta c^-6 \quad (4.23)$$

Obtenir la somme de ces deux équations nous donnera,

$$s^-(2+3+4)(c^-6 \frac{2}{6} - s^-6 \frac{2}{6}) = -s^-(2+3+4)c6 = \alpha s^-6 + \beta c^-6 \quad (4.24)$$

Alors, multiplier (4.20) par c^-6 nous donnera,

$$s^-(2+3+4)c^-6 \frac{2}{6} + c^-(2+3+4)s^-6c^-6 = \gamma c^-6 \quad (4.25)$$

Et multiplier (4.21) par $-s^-6$ nous donnera :

$$-c^-(2+3+4)c^-6s^-6 - s^-(2+3+4)s^-6 \frac{2}{6} = -\delta s^-6 \quad (4.26)$$

Obtenir la somme de ces deux équations nous donnera,

$$s^-(2+3+4)(c^-6 \frac{2}{6} - s^-6 \frac{2}{6}) = -s^-(2+3+4)c6 = \gamma c^-6 - \delta s^-6 \quad (4.27)$$

Nous pouvons soustraire les équations (4.24) et (4.27) pour obtenir le schéma simplifié suivant équation dans laquelle seul θ_6 est indéterminé :

$$\alpha s^-6 + \beta c^-6 = \gamma c^-6 - \delta s^-6 \rightarrow s^-6(\alpha + \delta) = c^-6(\gamma - \beta)$$

Réorganiser cette équation pour résoudre θ_6 nous donnera la formule suivante,

$$\theta_6 = 2\text{atan2}(\gamma - \beta, \alpha + \delta) \quad (4.28)$$

Il n'y a qu'une seule valeur pour θ_6 qui correspond à la rotation de l'articulation du Poignet 3.

4.3. Cinématique inverse par doubles quaternions

4.3.4 Transformation des cadres de l'épaule au poignet

Il ne nous reste plus que θ_2 , θ_3 et θ_4 à déterminer. Remarquez que leur les articulations correspondantes forment un manipulateur planaire 3R comme on le voit sur la figure ci-dessous,

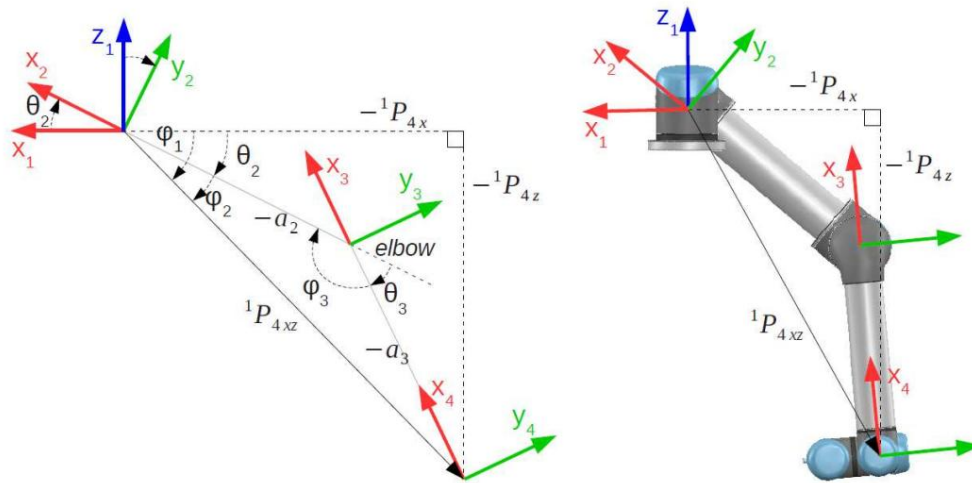


Figure 4.6 : Manipulateur planaire 3R formé par les articulations UR3 2 à 4 de [3].

Par conséquent, nous pouvons obtenir les produits de transformation M_i et N_i où ($2 \leq i \leq 4$). Cela nous conduira à 42 équations. L'équation des termes pour chaque produit de transformation correspondant nous laissera avec = 21 équations et $\frac{42}{2}$ nous essayons de trouver une relation entre certaines de ces équations et de les manipuler pour pouvoir résoudre explicitement nos angles d'articulation restants à déterminé.

Nous avons donc la représentation de quaternion dual suivante pour chaque articulation,

$$Q_2 = ([\bar{c}_2 - s_2j], < a_2c_2i + a_2s_2k >)$$

$$Q_3 = ([\bar{c}_3 - s_3j], < a_3c_3i - d_4j + a_3s_3k >)$$

$$Q_4 = ([\bar{c}_4 - s_4j], < 0 >)$$

et les inverses comme :

$$Q_2^{-1} = ([\bar{c}_2 + s_2j], < -a_2i >)$$

$$Q_3^{-1} = ([\bar{c}_3 + s_3j], < -a_3i + d_4j >)$$

$$Q_4^{-1} = ([\bar{c}_4 + s_4j], < 0 >)$$

Chapitre 4. Application des quaternions à Universal Robot UR3

Par conséquent, notre produit de transformation double quaternion M_i est formulé comme suit :

$$\begin{aligned} M_4 &= Q_4 \\ M_3 &= Q_3 \quad M_4 = Q_3 \quad Q_4 \\ M_2 &= Q_2 \quad M_3 = Q_2 \quad Q_3 \quad Q_4 \end{aligned}$$

Et le produit quaternionique N_i comme :

$$\begin{aligned} N_2 &= Q_2 \quad [w_4, < a_4, b_4, c_4 >, < x_4, y_4, z_4 >) \\ N_3 &= Q^{-1}_2 \quad N_2 = Q^{-1}_2 \quad Q_2 \quad 4 \\ N_4 &= Q^{-1}_3 \quad N_3 = Q^{-1}_3 \quad Q_2^{-1} \quad Q_4 \end{aligned}$$

Ces équations sont énumérées à l'annexe C.

En dehors de cela, nous devons également traduire nos coordonnées de base et nos coordonnées finales puisque nous travaillons uniquement du cadre 2 (épaule) au cadre 4 (poignet 1). Pour traduire les coordonnées, nous utilisons la même technique que nous avons utilisée pour déterminer les coordonnées de θ_1 ,

$$\begin{aligned} Q_{FK} &= Q_1 \quad Q_2 \quad Q_3 \quad Q_4 \quad Q_5 \quad Q_6 = Q_1 \quad 6 \quad (4,29) \\ Q^{-1}_1 \quad Q_{FK} &= Q_2 \quad Q_3 \quad Q_4 \quad Q_5 \quad Q_6 \quad = Q_2 \quad 6 \quad (4,30) \\ Q^{-1}_1 \quad Q_{FK} \quad Q^{-1}_6 &= Q_2 \quad Q_3 \quad Q_4 \quad Q_5 \quad = Q_2 \quad 5 \quad (4,31) \\ Q^{-1}_1 \quad Q_{FK} \quad Q^{-1}_6 \quad Q_5^{-1} &= Q_2 \quad Q_3 \quad Q_4 \quad = Q_2 \quad 4 \quad (4,32) \end{aligned}$$

Notez que nous pouvons déjà former complètement Q_1 , Q_5 et Q_6 . Pour Q_4 , nous n'avons qu'à inverser sa traduction. Pour ce faire, nous utilisons son inverse Q^{-1} de la même manière que nous l'avons fait dans la traduction précédente. Nous avons,

$$Q_4^{-1} = ([\bar{c}_5 + \bar{s}_5 k], < d_6 i >)$$

Et nous inversons la traduction par,

$$\begin{aligned} (Q_2 \quad 4 \quad Q^{-1}_4) p &= (Q_2 \quad Q_3 \quad Q_4 \quad Q^{-1}_4) p \quad (4.33) \\ &= (Q_2 \quad Q_3) p \\ &= (Q_2 \quad 3) p \end{aligned}$$

Nous obtenons les coordonnées par la partie traduction de ce produit,

$$\begin{aligned} (Q_2 \quad 4 \quad Q^{-1}_4) p &= < \frac{2}{4} \quad p_4^{-1} \quad (q_4^2)^{-1} + p \quad \frac{2}{4} > \quad (4.34) \\ &= < x_4 i + y_4 j + z_4 k > \end{aligned}$$

où $< x_4, y_4, z_4 >$ sont les coordonnées dans l'image 4 par rapport à l'image 2.

4.3. Cinématique inverse par doubles quaternions

Dérivation de l'angle d'articulation θ_3 (coude)

Pour dériver une formule pour θ_3 , nous utilisons les équations (C.19) et (C.21) dans la transformation de l'épaule au poignet 1 avec la valeur précédemment calculée coordonnées $\langle x_4, y_4, z_4 \rangle$.

$$\underline{M_{25} = N_{25}}$$

$$a_3 \cos(\theta_2 + \theta_3) + a_2 \cos \theta_2 = x_4 \quad (4.35)$$

$$\underline{M_{27} = N_{27}}$$

$$a_3 \sin(\theta_2 + \theta_3) + a_2 \sin \theta_2 = z_4 \quad (4.36)$$

La mise au carré de ces deux équations et leur addition nous donnera,

$$a_3^2 + 2a_2a_3 \cos \theta_3 + a_2^2 = x_4^2 + z_4^2 \quad (4.37)$$

Maintenant, nous pouvons calculer directement pour θ_3 par,

$$\theta_3 = \pm \arccos \frac{x_4^2 + z_4^2 - a_2^2 - a_3^2}{2a_2a_3} \quad (4.38)$$

On considère deux valeurs pour θ_3 qui correspond à la rotation de l'articulation du coude d'être en haut ou en bas.

Dérivation de l'angle d'articulation θ_2 (épaule)

Puisque nous avons déjà θ_3 , nous pouvons calculer pour θ_2 en utilisant les équations (C.12) et (C.14) également dans la Transformation Épaule au Poignet 1 en utilisant le même coordonnées.

$$\underline{M_{35} = N_{35}}$$

$$a_3 \cos \theta_3 = -a_2 + x_4 \cos \theta_2 + z_4 \sin \theta_2 \quad (4.39)$$

$$\underline{M_{37} = N_{37}}$$

$$a_3 \sin \theta_3 = -x_4 \sin \theta_2 + z_4 \cos \theta_2 \quad (4.40)$$

Et en utilisant des manipulations similaires, nous pouvons calculer directement pour θ_2 en,

$$\theta_2 = \text{atan2}(z_4(a_2 + a_3 \cos \theta_3) - x_4a_3 \sin \theta_3, x_4(a_2 + a_3 \cos \theta_3) + z_4a_3 \sin \theta_3) \quad (4.41)$$

Nous considérons une valeur pour θ_2 qui correspond à la rotation de l'épaule.

Dérivation de l'angle d'articulation θ_4 (poignet 1)

Étant donné θ_3 et θ_2 , nous pouvons calculer pour θ_4 en utilisant les équations (C.1) et (C.3) et la partie rotation de la transformation Q_2 donnée par $q_4 = w_4 + a_4i + b_4j + c_4k$.

$$\begin{aligned} & \underline{M_{41} = N_{41}} \\ & \text{parce que } \frac{\theta_4}{2} = w_4 \cos \frac{\theta_2 + \theta_3}{2} - b_4 \sin \frac{\theta_2 + \theta_3}{2} \end{aligned} \quad (4.42)$$

$$\begin{aligned} & \underline{M_{43} = N_{43}} \\ & - \sin \frac{\theta_4}{2} = w_4 \sin \frac{\theta_2 + \theta_3}{2} + b_4 \cos \frac{\theta_2 + \theta_3}{2} \end{aligned} \quad (4.43)$$

Nous pouvons simplement calculer pour θ_4 par,

$$\theta_4 = -2 \arctan \frac{w_4 \sin \frac{\theta_2 + \theta_3}{2} + b_4 \cos \frac{\theta_2 + \theta_3}{2}}{w_4 \cos \frac{\theta_2 + \theta_3}{2} - b_4 \sin \frac{\theta_2 + \theta_3}{2}} \quad (4.44)$$

Nous considérons une valeur pour θ_4 qui correspond à la rotation de l'articulation du Poignet 1.

4.3.5 Solutions d'angles de joint

L'utilisation des équations (4.11), (4.13), (4.28), (4.38), (4.41) et (4.44) nous permettra de calculer pour une combinaison d'angles ($\theta_1, \theta_2, \theta_3, \theta_4, \theta_5, \theta_6$) qui donnez-nous notre transformation souhaitée Q_1 . Chacun de nos angles communs de ces équations a le nombre suivant de valeurs possibles,

$$\begin{array}{ccc} \theta_1 \times 2 & \theta_2 \times 1 & \theta_3 \times 2 \\ \theta_4 \times 1 & \theta_5 \times 2 & \theta_6 \times 1 \end{array}$$

Par conséquent, il existe au total $2 \times 1 \times 2 \times 1 \times 2 \times 1 = 8$ combinaisons d'angles d'articulation possibles. Il est encore nécessaire de filtrer ces valeurs dans notre application de programmation pratique car il peut encore arriver que seules certaines de ces combinaisons nous donnent exactement la transformation souhaitée. Dans cette étude, cela se fait en utilisant le principe de la cinématique directe. Maintenant, nous avons notre entrée en tant qu'angles articulaires de notre combinaison actuelle et nous obtenons leur transformation correspondante,

$$Q_1(\theta_1) Q_2(\theta_2) Q_3(\theta_3) Q_4(\theta_4) Q_5(\theta_5) Q_6(\theta_6) = Q_{FK} \quad (4.45)$$

Nous vérifions si cette transformation est équivalente à notre transformation souhaitée où nous devons également tenir compte de quelques petits écarts dus à des erreurs numériques. Cette procédure est illustrée dans notre application de programmation.

4.4 Application de programmation pratique

L'application de programmation Python pour dériver l'avant et l'inverse

La cinématique de UR3 est présentée dans cette section.

4.4.1 Paramètres de Denavit-Hartenberg

Les paramètres Denavit-Hartenberg de UR3 sont donnés dans le tableau suivant :

UR3				
Kinematics	theta [rad]	a [m]	d [m]	alpha [rad]
Joint 1	0	0	0.1519	$\pi/2$
Joint 2	0	-0.24365	0	0
Joint 3	0	-0.21325	0	0
Joint 4	0	0	0.11235	$\pi/2$
Joint 5	0	0	0.08535	$-\pi/2$
Joint 6	0	0	0.0819	0

Tableau 4.2 : Paramètres de Denavit-Hartenberg pour UR3 de [20]

Ces paramètres sont décrits par le schéma suivant :

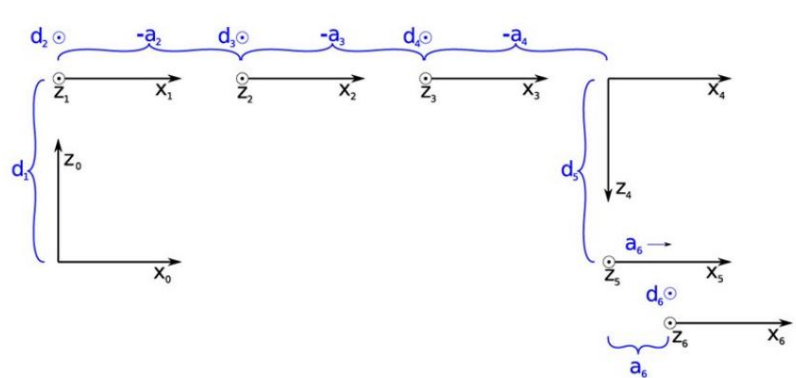


Figure 4.7 : Diagramme des paramètres DH de l'UR3 de [20]

La configuration des paramètres DH dans le programme suit l'ordre du tableau

4.1 en utilisant les valeurs données par le tableau 4.2.

```
1 # UR3 Denavit-Hartenberg Paramètres
2 ré = [ 0.1519, 0.101, 0.235, 0.0819, 0.08535,
3 un = [ 0, 0, 0.24365, -0.21325, 0, 0 ]
alpha a = [ 0, pi/2, 0, 0, pi/2, -pi/2 ]
```

5L'utilisation et la dérivation des paramètres de Denavit-Hartenberg peuvent être trouvées dans [23]

4.4.2 Application de programmation Python

Nous utilisons la bibliothèque python urx pour communiquer et contrôler le robot UR3.

```
1 urx d'importation
2 depuis urx import urrobot
3 rob = urx . Robot ( " 1 4 7. 2 2 9. 1 3 2. 2 4 9" ) #Adresse IP de UR3
```

L'adresse IP du robot UR3 peut être vue dans l'onglet "À propos" de l'UR3

Interface utilisateur graphique Polyscope.

Classe double quaternion

Nous définissons d'abord notre objet quaternion dual. Comme nous l'avons décrit, il consiste de deux quaternions où le premier quaternion représente la rotation et le deuxième quaternion représente la traduction. La bibliothèque pyquaternion est utilisé pour définir ces objets quaternion et nous avons créé une classe pour le Dual Quaternion suivant les propriétés que nous avons énumérées dans le chapitre sur les quaternions. Le la multiplication des quaternions doubles est également définie dans cette classe suivant la propriétés que nous avons énumérées dans le chapitre sur les quaternions doubles.

```
1 classe DualQuaternion :
2 # Initialise un double quaternion
3     attribut __en__ ( soi , q=Quaternion ( ) , p=0 ) :
4         soi . q = q
5         soi . p = p
6
7 # Multiplication de quaternions doubles
8     par défaut mult ( self , dq ) :
9         # Pour la partie rotation
10            q1 = soi . q
11            q2 = dq . q
12
13        # Pour la partie traduction
14            p1 = soi . p
15            p2 = dq . p
16
17        return DualQuaternion ( q1 q2 , q1 p2 q1 . conjugué e+p1 )
18
19 # Vérifie l'égalité des quaternions doubles
20     def égal s ( self , dq ) :
21         # Pour la partie rotation
22            q1 = soi . Q. vecteur
23            q2 = dq . Q. vecteur
24
25        # Pour la partie traduction
26            p1 = soi . p. vecteur
27            p2 = dq . p. vecteur
28
```

4.4. Application de programmation pratique

```

29 # Vérifie l'égalité
30 print ( "\nComparaison des quarts doubles : " )
31 imprimer ( " Rotation : , q1 , "
32 Translation : , p1 ,
33
34 si ( ( np . allclose ( p1 , p2 , atol=1e-03)
35 ou np. toutfermer ( p2 , p1 , atol=1e-03) )
36 et ( np . allclose ( q1 , q2 , atol=1e-03)
37 ou np. toutfermer ( q2 , q1 , atol=1e-03) ) ) :
38 retourner Vrai
39 retour F al se
40
41 # Transforme un point dans l'espace
42 def transform ( self , pose ) :
43 # Pose tournée
44 ro tP ose = self . Q. tourner (poser),
45 print ( "\n A f t e r R o t a t i o n : ro tP ose )
46
47 # Pose traduite
48 tran sP ose = soi . p. vecteur + ro tP ose
49 print ( "Après T r a n s l a t i o n : tran sP ose )
50
51 retour tran sP ose

```

Pour faire tourner un vecteur, nous avons utilisé la méthode pyquaternion rotate car elle normalise le quaternion avant d'appliquer la rotation. Cela se fait dans le codes suivants dans la bibliothèque pyquaternion :

```

1 def rotation ( self,vector ) :
2     Faire pivoter un vecteur 3D par la rotation stockée dans le
   Objet quaternion .
3
4     Paramètres :
5         vecteur : un vecteur à 3 vecteurs spécifié comme une séquence ordonnée
   de 3 nombres réels correspon di ngto x et z v al ues . , y ,
6         Certains types reconnus sedare : numpy arrays
   , listes et tuples.
7         Un 3-vecteur peut aussi être représenté par un
   Objet Quaternion who ' sscalarpartis 0 et vectorpartis
   le 3-vecteur rouge requis .
8         Il est donc possible d'appeler ' Quaternion . tourner (q
   ) ' avec un o le rquater ni sur l'objet comme entrée .
9
10    Retour :
11        Le vecteur pivoté est retourné avec le même type qu'il était
   entréespécifiée.
12
13    R a i s e s :
14        TypeError : si l'un des éléments vectoriels ne peut pas être
   converti en nombre réel .

```

Chapitre 4. Application des quaternions à Universal Robot UR3

```

15 Erreur de valeur : if ' vecteur ' ne peut être interprété comme
3-v ectorora objet Quaternion .
16
17
18 ifinstance ( vecteur , Quaternion ) :
19     retourner soi-même. rotationquaternion ( vecteur )
20 q = Quaternion ( vecteur=vecteur )
21 a = soi. tourner quaternion ( q ) . vecteur
22 ifinstance ( liste de vecteurs ) :
23     l = [ x pour x dans un ]
24     retour l
25 elifinstance ( vectorl = [ x , tu pl e ) :
26     pour x dans a ]
27     retourtu pl e ( l )
28 sinon :
29     retourner un
30
31 def rotationquaternion ( self , q ) :
32     Faites pivoter l'eau sans vecteur en utilisant la rotation stockée.
33
34     Paramètres :
35     q : Le vecteur à faire pivoter , inquater ni sur la forme ( 0 +
xi + yj + kz )
36
37     Retour :
38     Un objet Quaternion représentant le vecteur tourné
inquater,,ni à partir de ( 0 + xi + yj + kz )
39
40     soi . no-rm ali se ( )
41     retourner soi q soi . conjuguer
42
43     def no-rm ali se ( self ) :
44         """ Ob jec tis gua r an te d'être au ni tquater ni après
appelant ça
45         opéra ti on À MOINS QUE l'objectiseq ui v al entto Quaternion
( 0 ) """
46
47     sinon soi. estunité (-) :
48         n = soi. norme
49         si n > 0 :
50             soi . q = soi. q / n

```

On note aussi que dans le code précédent, on a défini comment déterminer si deux les quaternions doubles sont à peu près égaux. Nous avons utilisé la méthode allclose de la bibliothèque numpy qui vérifie si deux tableaux sont égaux au niveau des éléments dans un tolérance. Nous le faisons pour la cinématique inverse lorsque nous voulons filtrer les combinaisons d'angles résultantes par leurs transformations correspondantes et il faut tenir compte des erreurs numériques qui peuvent survenir lors des calculs.

4.4. Application de programmation pratique

Cinématique avant

Pour la cinématique directe, notre objectif est de déterminer la transformation finale (orientation et position) de l'effecteur du robot compte tenu de ses articulations' rotations angulaires. Tout d'abord, nous devons obtenir les angles d'articulation actuels de notre robot. Pour ce faire, nous utilisons la méthode `getj` de la bibliothèque `urx`.

```
1 # Obtient les vues jointes actuelles
```

```
2 jangles = voler . obtenirj ( )
```

Cela renvoie la valeur des six angles d'articulation à partir de la base jusqu'au poignet

3. Nous définissons également deux approches pour dériver un quaternion dual. L'un est par la représentation axe-angle (AA) et par la matrice de transformation (TM).

Nous définissons différentes coordonnées TCP initiales pour chacune de ces approches.

```
1 # Définir la position initiale de TCP dans l'espace en fonction de la valeur correspondante
```

```
cadre
```

```
2 TCP_endf = np . tableau ( [ 0 3 TCP_basef = np . , 0 , 0 . 2 ] ) # basé sur l'image de fin ( 0 , 0 , 1 )
tableau ( [ 0 - , -0 . 2 , 0 ] ) # basé sur la trame de base ( 0 , -1 , 0 )
```

La raison en est que pour la première approche qui est par AA, nous devons considérer les coordonnées TCP par rapport à la trame de base. Et pour la TM approche, nous devons considérer les coordonnées TCP par rapport à sa trame.

La méthode `get forward kinematics` appelle maintenant la dérivation du dual quaternions pour chacune des six articulations en fonction de l'approche choisie et calcule le quaternion double final représentant la transformation finale en multipliant suivant l'ordre de transformation (joints).

```
1 # Obtient la position actuelle en utilisant Forward Kinematics
```

```
2 def getforward kinematics ( ans , jangles ) :
```

```
3
4     print ( "\nAngles de jonction actuels : " , cliquetis)
```

```
6 # Dérive les quarts doubles correspondants pour chaque joint
```

```
à partir de leur représentation Angle-Axe
```

```
si ( répond == "AA" ) :
```

```
    print ( "\nDérivation de la cinématique directe à partir de Angle-Axe
    Représentation . . . " )
```

```
9     DualQuatList = quat_dérivé_dérivé de AA ( jangles )
dix     pose initiale = TCP_basef
```

```
11 # Dérive les quarts doubles correspondants pour chaque articulation
```

```
à partir de leurs Transformations Matricielles
```

```
12 elif ( ans == "TM" ) :
```

```
13     print ( "\nDérivation de la cinématique directe à partir de la Transformations
    Matricielles . . . " )
```

```
14     DualQuatList = dérive le dual Quaternion de TM ( jangles )
15     pose initiale = TCP_endf
```

```
16 sinon :
```

```
17     print ( "Entrée erronée !" )
```

Chapitre 4. Application des quaternions à Universal Robot UR3

```

18         retour
19
20 # Calcule le Quaternion Dual Final en multipliant tous les
    Double Quaternions
21     FKDualQuat = DualQuaternion ( )
22     pour dq dans DualQuatList :
23         FKDualQuat = FKDualQuat . mult ( dq )
24
25     print ( "\nForward Kinematics ( by Dual Quaternion ) : \n ( [ "
        FKDualQuat . q , " ] , , FKDualQuat . p , "> )" )
26
27 # Transformer la pose initiale par Quaternion-Vecteur Multiplication
28     print ( "\nCoordonnées TCP initiales ( atzeropose ) : initialpose )
        -
29     currpose = FKDualQuat . transformer (pose initiale)
        -
30     print ( "\nTransformed TCP Coordinates: \n", np . round ( currpose
        , 3 ) )
31
32     retour FKDualQuat

```

La sortie de cette méthode est la pose actuelle ou les coordonnées TCP de
le robot après qu'il a été transformé (rotation et translation) et avec
par rapport au cadre de base.

Quaternions doubles de la représentation Axis-Angle

La dérivation de la cinématique directe par l'approche Axis-Angle est
fait dans la méthode dériver le double quat de AA :

```

1 # Dériver le double quaternion à partir de l'angle et de l'axe de rotation
2 def derive_double_quatfrom AA (jangles) :
3 # Récupère les joints à partir de la position actuelle
4     theta = jangles
5
6 # Définir la traduction pour chaque articulation
7     trans = ( [[ 0 ré [ 0. ] , 0 [ une [ 2 ]
8                 cos ( theta [ 1 ] ) , [ une 0 , un [ 2 ] sin ( theta [ 1 ] ) ] , #P1
9                 [ 3 ] cos ( theta [ 2 ] ) , [ ré -d [ 3 ] une [ 3 ] sin ( theta [ 2 ] ) ] , #P3
10                [ 4 ] sin ( theta [ 3 ] ) , [-d 0 , -d [ 4 ] cos ( theta [ 3 ] ) ] , #P4
11                [ 5 ] sin ( theta [ 4 ] ) , [ 0 -d [ 5 ] cos ( theta [ 4 ] ) , 0 ] , #P5
12                0 ] ] , # P6
13
14 # Build the Dual Quaternion
15     DQListe = [ ]
16     al p = 0
17
18     pour moi in range ( 6 ) :
19         # Build the dual quaternion
20         qscalaire = cos ( theta [ i ] / 2 )

```

4.4. Application de programmation pratique

```

21     al p += alpha une [ je ]
22     qvecteur = np . m ul ti pl é ( sin ( thêta [ i ] / 2 ) , al p ) )      [ 0 ,  -sin ( al p ) ,      parce que (
23
24     quatRot = np . concaténer ( ( [ qscalaire ] , qvecteur ) )
25
26     DualQuat = DualQuaternion ( Quaternion ( quatRot ) , Quaternion (
    vecteur=trans [ je ] ) )
26     print ( "Quaternion double : ( [ "      , DualQuat . q , " ] , , DualQuat . p , ">"
    " ) )
27
28     # Stocker le Dual Quaternion dans la liste
29     DQListe . ajouter (DualQuat)
30
31     retourner la liste DQ

```

Pour calculer la transformation finale, nous définissons d'abord la translation pour chaque articulations. Cela correspond à la partie traduction de nos quaternions duaux dans laquelle on utilise les paramètres de Denavit-Hartenberg (a , d , θ). Ensuite nous construisons les quaternions de rotation pour les six articulations en utilisant les angles du robot et son quaternion de traduction correspondant par les traductions dérivées. Le la variable alp suit le paramètre α dans les paramètres DH et nous utilisons ceci dans la détermination de l'axe de rotation. Cela peut avoir des valeurs 0 ou qui $\frac{\pi}{2}$ rendra l'un ou l'autre des termes $-\sin(alp)$ et $\cos(alp)$ nul ou ± 1 . Alors, nous créons l'objet quaternion double en combinant ces deux quaternions. Enfin, nous stockons ces quaternions doubles dans une liste et la renvoyons au précédent méthode qui effectue la transformation.

Quaternions doubles de la matrice de transformation

Une autre approche pour dériver la cinématique vers l'avant peut être par une transformation matrice. Ceci est fait dans la méthode dérivée du double quat de TM. L'habituel la convention de dérivation de la cinématique avant consiste à utiliser des matrices de rotation et vecteurs de translation qui sont combinés pour former une matrice de transformation (4x4). A partir de cette matrice de transformation, nous pouvons la convertir en un quaternion double en convertissant les matrices de rotation (3x3) en quaternions unitaires et les vecteurs de translation (3x1) en un quaternion pur. La matrice de rotation en quaternion la conversion a été abordée à la section 3.4.2. Pour la matrice de transformation, nous suivre sa dérivation dans [3] et [16].

```

1 # Obtient le Dual Quaternion pour chaque articulation
2 def de rive Du al Qu a te rni on from TM ( jangles ) :
3     print ( "Computing T r an sfo rm a ti on Matrix par Denavit-Har tenbe rg
    Paramètres . . . " )
4 # Récupère les joints à partir de la position actuelle
5     thêta = jangles
6 # Obtient les matrices de transformation pour chaque 6 trames

```

Chapitre 4. Application des quaternions à Universal Robot UR3

```

7 T = np . zéros ( forme = ( 6
                                , 4 , 4 ) )
8
9 # B uil ds le Dual Qu a te m i on s
dix DQListe = [ ]
11
12 pour moi in range ( 6 ) :
13     T[ je ] = ( [ [ cos ( thème [ je ] ) , -sin ( thème [ je ] ) , une [ je ] ] [ pécRé ( thème
14                 [ je ] ) cos ( alpha une [ je ] ) , cos ( thème [ je ] ) cos (
15     alph une [ je ] ) , -sin ( alph une [ je ] ) , -sin ( alph une [ je ] ) d [ je ] ] ,
16                 [ sin ( thème [ je ] ) sin ( alph une [ je ] ) , cos ( thème [ je ] ) sin (
17     alpha a [ je ] ) , cos ( alpha une [ je ] ) , cos ( alpha une [ je ] ) d [ je ] ] ,
18                 [ 0 , 0 0 , , 1 ] ] )
19
20     print ( "\ n T r an sfo rm a ti on Matrix for Frame" , je , " à " , je +1, "\ n"
21     , T[ je ] )
22
23     # D é ri ve le Quaternion Double de la T r an sfo rm a ti on
24     Matrice
25     DualQuat = DualQuaternion ( Quaternion ( matrice = T[ i ] ) ,
26     Quaternion ( vecteur = T[ i ] [ 0 : 3
27     , 3 ] ) )
28     print ( "Quaternion double : ( [ " , DualQuat . q , " ] , , Duā"Quat . p , ">"
29     " ) )
30
31     # Stocker le Dual Quaternion dans la liste
32     DQListe . ajouter (DualQuat)
33
34 retourner la liste DQ

```

Comme dans l'approche précédente, nous stockons les quaternions doubles dans une liste et passez-le à la première méthode qui effectue la transformation. Le dual final quaternion représentant la transformation finale sera différent du autre approche. En effet, dans la dérivation par AA, nous basons la quaternions doubles uniquement sur le cadre de base tandis que pour l'approche TM, chacun les quaternions doubles sont dérivés par rapport à leurs propres cadres. Dans n'importe lequel de ces approches, les coordonnées TCP finales seront toujours les mêmes.

Application pratique

Pour l'application, nous utilisons une méthode record pose qui prend le courant pose du robot et l'enregistre dans un fichier texte. Les coefficients de ce dual le quaternion dérivé des angles d'articulation du robot est ensuite stocké dans un texte déposer. Plusieurs poses peuvent être enregistrées dans le fichier texte. A part cela, le la cinématique avant est également utilisée pour filtrer les combinaisons d'angles dérivées dans les cinématiques inverses en vérifiant si leurs transformations correspondent aux transformation souhaitée écrite dans le fichier texte.

4.4. Application de programmation pratique

Cinématique inverse

Pour la cinématique inverse, notre but est de déterminer les angles articulaires qui donnent la pose souhaitée (rotation et translation). Ainsi, on nous donne maintenant la transformation finale et nous l'utilisons pour dériver les valeurs des six angles articulaires correspondants. Nous utilisons la méthode d'obtention de la cinématique inverse pour effectuer la dérivation. Nous commençons par définir les coordonnées à l'image 6 (poignet 3) et les coordonnées correspondantes à l'image 5 (poignet 2).

```

1 # Sorties jointes à partir de cinématiques inverses
2 def getinversekinematics ( pose souhaitée ) :
3     print ( "\nDe si red Pose : pose souhaitée . q , pose souhaitée . p );
4 # Transformation désirée de Frame 1 à 6
5 DQ16 = pose souhaitée
6 # Rotation Part
7     w6 = DQ16. q [ 0 ]
8     a6 = DQ16. q [ 1 ]
9     b6 = DQ16. q [ 2 ]
10    c6 = DQ16. q [ 3 ]
11 # Translation Partie
12    x6 = DQ16. p [ 1 ] ;
13    y6 = DQ16. p [ 2 ] ;
14    z6 = DQ16. p [ 3 ] ;
15
16 # Supprimer la traduction du cadre 5
17    DQ5i = DualQuaternion ( p=Quaternion ( vecteur =[0 , d [ 5 ] , 0 ] ) )
18    DQ15p = DQ16. multiple ( DQ5i )
19    x5 = DQ15p . p [ 1 ]
20    y5 = DQ15p . p [ 2 ]
21    z5 = DQ15p . p [ 3 ]

```

La traduction du poignet 3 au poignet 2 a été démontrée dans la section 4.3.3.

Calcul pour Base (θ_1)

Pour calculer pour θ_1 , nous avons utilisé l'équation (4.11) :

$$\theta_1 = \text{atan2}(-x_5, y_5) \pm \text{atan2}\left(x_5^2 + y_5^2 - d_4^2, -d_4\right)$$

Ceci est exprimé par le code suivant,

```

1 # Calculer les valeurs possibles pour Theta 1
2     theta 1 =[]
3     t 1 1 = atan2 (-x5 , y5)+atan2 ( sqrt ( x5^2+y5^2-d [ 3 ]^2 ) ,-d [ 3 ] )
4     t 1 2 = atan2 (-x5 , y5)-atan2 ( sqrt ( x5^2+y5^2-d [ 3 ]^2 ) ,-d [ 3 ] )
5 # ( Base ) Theta 1 : 2 Valeurs
6     theta 1 . ajouter ( t 1 1 )
7     theta 1 . ajouter ( t 1 2 )

```

Cela donne au plus deux valeurs possibles pour θ_1 .

Chapitre 4. Application des quaternions à Universal Robot UR3

Calcul pour le poignet 2 (θ_5)

Pour calculer pour θ_5 , nous avons utilisé l'équation (4.13) :

$$\theta_5 = \pm \arccos \frac{x_6 \sin \theta_1 - y_6 \cos \theta_1 - d_4}{d_6}$$

Ceci est exprimé par le code suivant,

```

1 # Calculer les valeurs possibles pour Theta 5
2   theta_1_5 = []
3   pour t_1 dans theta_1 : # Pour chaque valeurs de theta_1
4     # Evaluate l'expression pour Theta 5
5     exp5 = ( x6 * sin ( t_1 ) - y6 * cos ( t_1 ) - d [ 3 ] ) / d [ 5 ]
6     # Vérifie si dans le domaine for arccos
7     si ( exp5 <= 1 et exp5 >= -1 ) :
8       t_5 = acos ( exp5 )
9     elif ( rond ( exp5 ) <= 1 et rond ( exp5 ) >= -1 ):
10      t_5 = acos ( rond ( exp5 ) )
11     sinon : continuer
12   # ( Poignet 2 ) Theta 5 : 2 Valeurs
13   theta_1_5 . ajouter ( [ t_1 t_5 ] )
14   theta_1_5 . ajouter ( [ t_1 , -t_5 ] )

```

Nous considérons que $\arccos$ a un domaine de seulement $[-1, 1]$. Par conséquent, pour empêcher erreurs d'exécution, nous arrondissons les valeurs et vérifions si elles sont maintenant à l'intérieur de ce domaine. Cela donne à nouveau deux valeurs possibles pour θ_5 .

Calcul pour le poignet 3 (θ_6)

Pour calculer pour θ_6 , nous avons utilisé l'équation (4.28) :

$$\theta_6 = 2 \arctan 2(\gamma - \beta, \alpha + \delta)$$

où,

$$\alpha = \frac{wc^{-1} + cs^{-1}}{c^{-5}} \quad \beta = \frac{ac^{-1} + bs^{-1}}{s^{-5}} \quad \gamma = \frac{comme^{-1} - bc^{-1}}{c^{-5}} \quad \delta = \frac{ws^{-1} - cc^{-1}}{s^{-5}}$$

Ceci est exprimé par le code suivant,

```

1 # Calculer les valeurs possibles pour Theta 6
2   theta_1_5_6 = []
3   pour t_1 t_5 en theta_1_5 :
4     t_1 = t_1 t_5 [ 0 ]
5     t_5 = t_1 t_5 [ 1 ]
6
7   # Évite la singularité pour Theta 6
8   si ( t_5 == 0 ) :

```

4.4. Application de programmation pratique

```

9      t156 = [ t1 si , t5 , 0 ]
dix    t156 pas en theta 156 :
11      theta 156 . ajouter ( t156 )
12      continuer
13
14      # Evalué l'expression si onfor Theta 6
15      tm1 = ( ( w6 cos ( t1 / 2 ) + c6 sin ( t1 / 2 ) ) / cos ( t5 / 2 ) )
16      tm2 = ( ( a6 cos ( t1 / 2 ) + b6 sin ( t1 / 2 ) ) / sin ( t5 / 2 ) )
17      tm3 = ( ( a6 sin ( t1 / 2 ) - b6 cos ( t1 / 2 ) ) / cos ( t5 / 2 ) )
18      tm4 = ( ( w6 sin ( t1 / 2 ) - c6 cos ( t1 / 2 ) ) / sin ( t5 / 2 ) )
19
20      # ( Poignet 3 ) Thêta 6 : 1 Valeur
21      t6 = 2 atan2 ( tm3 - tm2 tm1 + tm4 )
22      theta 156 . ajouter ( [ t1 t6 ] ) , t5 ,

```

Cela ne donne qu'une seule valeur possible pour θ_6 .

Maintenant, afin de définir les angles suivants. Nous devons effectuer une traduction encore. Nous devons obtenir la transformation du cadre 2 (épaule) au cadre 4 (poignet 1), donc nous avons besoin de Q_2 . Ceci a été discuté dans la section 4.3.4.

```

1 # Dériver la transformation de la trame 2 à la trame 4
2      DQ24f = [ ]
3      pour t156 en theta 156 :
4          t1 = t156 [ 0 ]
5          t5 = t156 [ 1 ]
6          t6 = t156 [ 2 ]
7          # Définir l'Inverse Dual Quaternion s Joints 1 6 , 5 ,
8          q1 = np . concatener ( ( [ cos ( t1 / 2 ) ] , np . multiplier ( sin ( t1 / 2 ) ,
9              [ 0 , 0 , -1 ] ) ) )
10         DQ1i = DualQuaternion ( Quaternion ( q1 ) , Quaternion ( vecteur
11             =[ 0 , 0 , -d [ 0 ] ] ) )
dix    q5 = np . concatener ( ( [ cos ( t5 / 2 ) ] , np . multiplier ( sin ( t5 / 2 ) ,
12             [ 0 , 0 , 1 ] ) ) )
13         DQ5i = DualQuaternion ( Quaternion ( q5 ) , Quaternion ( vecteur =[ 0 ,
14             d [ 5 ] , 0 ] ) )
15         q6 = np . concatener ( ( [ cos ( t6 / 2 ) ] , np . multiplier ( sin ( t6 / 2 ) ,
16             [ 0 , 1 , 0 ] ) ) )
17         DQ6i = DualQuaternion ( Quaternion ( q6 ) , Quaternion ( vecteur
18             =[ 0 , 0 , 0 ] ) )
19         # Définir la Transformation from Frame 2 to Frame 6
20         DQ26 = DQ1i . multiple ( DQ16 )
21         # Définir la Transformation De la trame 2 à la trame 5
22         DQ25 = DQ26 . multiple ( DQ6i )
23         # Définir la Transformation de la trame 2 à la trame 4
24         DQ24 = DQ25 . multiple ( DQ5i )
25         # Supprimer la traduction du cadre 4
26         DQ4i = DualQuaternion ( p=Quaternion ( vecteur =[ 0 DQ24f . append , 0 , d [ 4 ] ] ) )
27         ( [ DQ24 . mult ( DQ4i ) , t156 ] )

```

Chapitre 4. Application des quaternions à Universal Robot UR3

Pour chaque combinaison d'angles de θ_1 , θ_5 et θ_6 , nous dérivons leurs double quaternion inverse Q^{-1}_1 , Q^{-1}_5 et Q^{-1}_6 . Nous utilisons ces inverses pour dériver la métamorphose $Q_2 \rightarrow Q_4$. Ensuite, nous inversons la traduction faite par Q_4 par aussi en utilisant son inverse Q_4^{-1} . Les transformations sont ensuite stockées dans une liste le long avec l'ensemble des angles $[\theta_1, \theta_5, \theta_6]$ qui les a générés. Tout cela sera utilisé dans le calcul des angles d'articulation restants.

Calcul pour le coude (θ_3)

Pour calculer pour θ_3 , nous avons utilisé l'équation (4.38) :

$$\theta_3 = \pm \arccos \frac{x_4^2 + z_4^2 - u_{32}^2 - u_n^2}{2a_2a_3}$$

Ceci est exprimé par le code suivant,

```

1 # Calculer les valeurs possibles pour Theta 3
2 le ta 1 3 5 6 q =[]
3 pour dq dans DQ24f :
4     w4 = dq [ 0 ] . q [ 0 ]
5     b4 = dq [ 0 ] . q [ 2 ]
6     x4 = dq [ 0 ] . p [ 1 ]
7     y4 = dq [ 0 ] . p [ 2 ]
8     z4 = dq [ 0 ] . p [ 3 ]
9     q4 = [ w4 b4 , x4 , y4 , z4 ]
10
11     t 1 = dq [ 1 ] [ 0 ]
12     t 5 = dq [ 1 ] [ 1 ]
13     t 6 = dq [ 1 ] [ 2 ]
14
15 # Evalu a te expres si onfor Theta 3
16 exp3 = ( ( x4^2 + z4^2 - a [3]^2 - a [2]^2 ) / ( 2 * une [2] * une [3] ) )
17
18 # Vérifie si dans le domaine forarccos
19 si ( exp3 <= 1 et exp3 >= -1 ) :
20     t 3 = acos ( exp3 )
21 elif ( rond ( exp3 ) <= 1 et rond ( exp3 ) >= -1 ) :
22     t 3 = acos ( rond ( exp3 ) )
23 sinon : continuer
24
25 # ( Coude ) Thêta 3 : 2 Valeurs
26 le ta 1 3 5 6 q . ajouter ( [ t1 t5 le t3 t5 , q4 ] )
27 6 q . ajouter ( [ t1 , -t3 , t5 , t6 , q4 ] )

```

Idem avec θ_5 , nous considérons également que les résultats sont à l'intérieur du domaine d'arccos. Cela donne au plus deux valeurs possibles pour θ_3 .

4.4. Application de programmation pratique

Calcul pour l'épaule (θ_2)

Pour calculer pour θ_2 , nous avons utilisé l'équation (4.41) :

$$\theta_2 = \text{atan2}(z_4(a_2 + a_3 \cos \theta_3) - x_4 a_3 \sin \theta_3, x_4(a_2 + a_3 \cos \theta_3) + z_4 a_3 \sin \theta_3)$$

Ceci est exprimé par le code suivant,

```

1 # Calculer les valeurs possibles pour Theta 2
2   le ta 1 2 3 5 6 q =[]
3   pour t1356q dans le ta 1 3 5 6 q :
4       t 1 = t1356q [ 0 ]
5       t 3 = t1356q [ 1 ]
6       t 5 = t1356q [ 2 ]
7       t 6 = t1356q [ 3 ]
8       x4 = t1356q [ 4 ] [ 2 ]
9       z4 = t1356q [ 4 ] [ 4 ]
dix  # Ev alu a te expres si onfor Theta 2
11      t 2 = atan2 ( ( z4 ( une [ 2 ] + une [ 3 ] cos ( t 3 ) ) - x4 a [ 3 ] sin ( t 3 ) ) , ( x4 ( une
      [ 2 ] + une [ 3 ] cos ( t 3 ) ) + z4 a [ 3 ] sin ( t 3 ) ) )
12      # ( Épaule ) Thêta 2 : 1 Valeur
13      le ta 1 2 3 5 6 q . ajouter ( [ t1 t5 , t2 , t3 , , t6 , t1356q [ 4 ] ] )

```

Cela donne une seule valeur possible pour θ_2 .

Calcul pour le poignet 1 (θ_4)

Pour calculer pour θ_4 , nous avons utilisé l'équation (4.44) :

$$\theta_4 = -2\text{atan2}(w_4 s^-(2+3) + b_4 c^-(2+3), w_4 c^-(2+3) - b_4 s^-(2+3))$$

Ceci est exprimé par le code suivant,

```

1 # Calculer les valeurs possibles pour Theta 4
2   le ta 1 2 3 4 5 6 =[]
3   pour t12356q dans le ta 1 2 3 5 6 q :
4       t 1 = t12356q [ 0 ]
5       t 2 = t12356q [ 1 ]
6       t 3 = t12356q [ 2 ]
7       t 5 = t12356q [ 3 ]
8       t 6 = t12356q [ 4 ]
9       w4 = t12356q [ 5 ] [ 0 ]
dix  b4 = t12356q [ 5 ] [ 1 ]
11      # Ev alu a te expres si onfor Theta 4
12      t 4 = -2 atan2 ( ( w4 sin ( ( t 2 + t 3 ) / 2 ) ) + ( b4 cos ( ( t 2 + t 3 ) / 2 ) ) , ( w4
      cos ( ( t 2 + t 3 ) / 2 ) ) - ( b4 sin ( ( t 2 + t 3 ) / 2 ) ) )
13      # ( Poignet 1 ) Thêta 4 : 1 Valeur
14      le ta 1 2 3 4 5 6 . ajouter ( [ t1 t3 , t2 , , t4 , t5 , t 6 ] )

```

Cela donne une seule valeur possible pour θ_4 .

Chapitre 4. Application des quaternions à Universal Robot UR3

Filtration des combinaisons d'angles et application pratique

Pour filtrer et vérifier les combinaisons d'angles résultantes, nous dérivons leur cinématique avant correspondante qui représente la transformation finale effectuée par ces angles. Si cette transformation n'est pas approximativement égale à notre transformation souhaitée, puis nous filtrons ces angles et ne laissons que ceux qui satisfont notre égalité dans la tolérance de $1e-3$. Ceci est fait ci-dessous.

```

1 # Obtenir le double quaternion final correspondant en fonction de la
   combinaison d'angles
2 jangles = []
3 print ( "\nSet of Possible Joint Angles : \n" for theta in the , le ta 1 2 3 4 5 6 )
4 ta 1 2 3 4 5 6 :
5     FinalDQ = getforward kinematics ( "AA" , theta )
6
7     print ( "\nComparaison des quaternions doubles : " )
8     imprimer ( " Pose souhaitée : \n ( [ " , DQ16. q , " ] , , DQ16. p , ">" )
9     print ( " Pose actuelle : \n ( [ " , FinalDQ . q , " ] , , FinalDQ . p , ">" )
10
11     if ( DQ16. equals ( FinalDQ ) ) :
12         print ( "Critères PASSÉS !" )
13         jangles.ajouter ( theta )
14     sinon : print ( "Critères ECHEC !" )
15
16     print ( "\nFinal Joint Angles : " , cliquetis )
17
18     cliquetis de retour

```

Pour l'application, nous construisons d'abord le quaternion dual représenté par les coefficients du fichier texte généré par la méthode record pose. Alors, nous appliquons la cinématique inverse et utilisons les premiers angles d'articulation résultants comme entrée pour le mouvement du robot. (Un critère supplémentaire peut également être ajouté afin de trouver la combinaison d'angles d'articulation optimale)

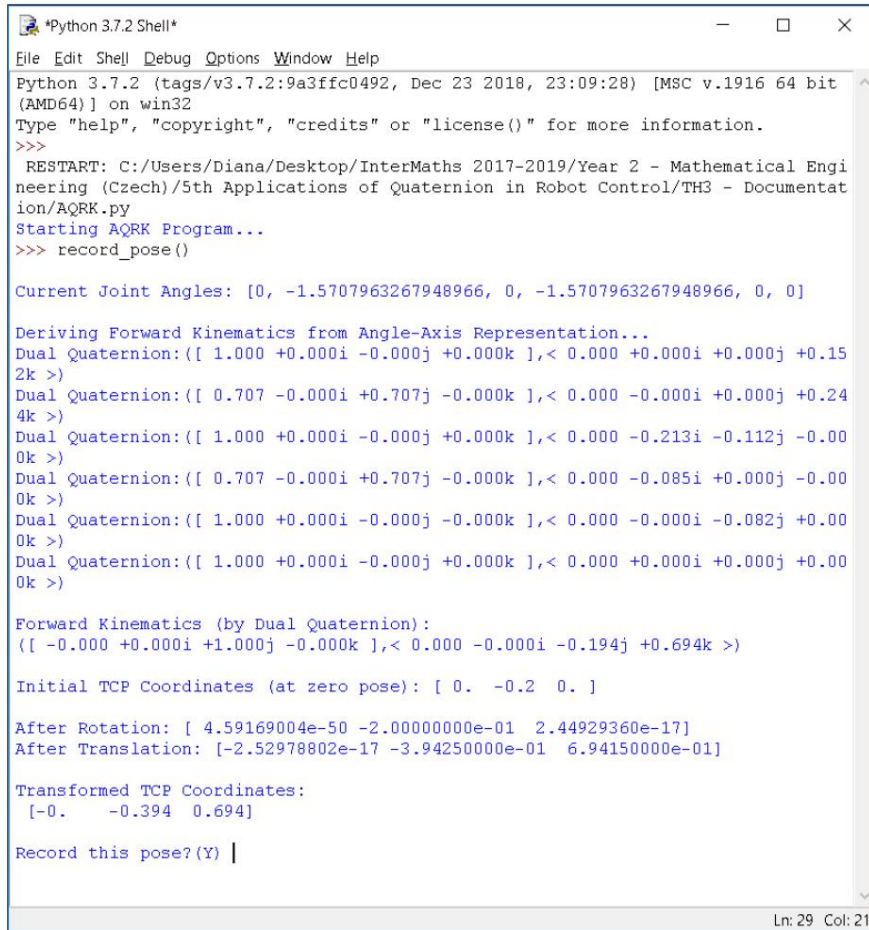
```

1 # Déplacer le robot d'une Transformation Dual Quaternion donnée
2 def déplacer robot ( dqcoeff ) :
3     # Prend les coefficients du quaternion dual et crée le
       dual quaternion sur
4     DQ de si redpose = DualQuaternion ( Quaternion ( array=dqcoeff
       [ 0 : 4 ] ) , Quaternion ( array=dqcoeff [ 4 : 8 ] ) )
5
6     # Applique l'Inverse Kinematics et retourne les angles joints
7     jointangles = getinversekinematics ( DQ de si redpose )
8
9     print ( "\nIl y a " , len ( angles articulaires ) , " poses possibles !" )
10
11     # Déplace le robot compte tenu des articulations
12     Rob . movej ( jointsangles [ 0 ] acc = 1 , v el = 1 , w ai t = F al se )

```

4.5 Vérification des résultats

Le programme commence par la fonction d'enregistrement de pose :



```
*Python 3.7.2 Shell*
File Edit Shell Debug Options Window Help
Python 3.7.2 (tags/v3.7.2:9a3ffc0492, Dec 23 2018, 23:09:28) [MSC v.1916 64 bit
(AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
RESTART: C:/Users/Diana/Desktop/InterMaths 2017-2019/Year 2 - Mathematical Engi
neering (Czech)/5th Applications of Quaternion in Robot Control/TH3 - Documentat
ion/AQRK.py
Starting AQRK Program...
>>> record_pose()

Current Joint Angles: [0, -1.5707963267948966, 0, -1.5707963267948966, 0, 0]

Deriving Forward Kinematics from Angle-Axis Representation...
Dual Quaternion:([ 1.000 +0.000i -0.000j +0.000k ],< 0.000 +0.000i +0.000j +0.15
2k >)
Dual Quaternion:([ 0.707 -0.000i +0.707j -0.000k ],< 0.000 -0.000i +0.000j +0.24
4k >)
Dual Quaternion:([ 1.000 +0.000i -0.000j +0.000k ],< 0.000 -0.213i -0.112j -0.00
0k >)
Dual Quaternion:([ 0.707 -0.000i +0.707j -0.000k ],< 0.000 -0.085i +0.000j -0.00
0k >)
Dual Quaternion:([ 1.000 +0.000i -0.000j -0.000k ],< 0.000 -0.000i -0.082j +0.00
0k >)
Dual Quaternion:([ 1.000 +0.000i -0.000j +0.000k ],< 0.000 +0.000i +0.000j +0.00
0k >)

Forward Kinematics (by Dual Quaternion):
([ -0.000 +0.000i +1.000j -0.000k ],< 0.000 -0.000i -0.194j +0.694k >)

Initial TCP Coordinates (at zero pose): [ 0. -0.2 0. ]

After Rotation: [ 4.59169004e-50 -2.00000000e-01 2.44929360e-17]
After Translation: [-2.52978802e-17 -3.94250000e-01 6.94150000e-01]

Transformed TCP Coordinates:
[-0. -0.394 0.694]

Record this pose?(Y) |
```

Il s'agit de la pose de base du robot donnée par l'ensemble d'angles suivant :

$$\begin{array}{lll}
 \text{Base : } \theta_1 = 0 & \text{Épaule : } \theta_2 = \frac{-\pi}{2} & \text{Coude : } \theta_3 = 0 \\
 \text{Poignet 1 : } \theta_4 = \frac{-\pi}{2} & \text{Poignet 2 : } \theta_5 = 0 & \text{Poignet3 : } \theta_6 = 0
 \end{array}$$

Les quaternions doubles ci-dessus sont dérivés par rapport à ces angles avec leur axe de rotation et leurs translations correspondantes. Ceci peut être vérifié dans la section 4.2.1. Ceux-ci sont ensuite multipliés et le produit est défini comme le quaternion double représentant la transformation finale. Le point central de l'outil de l'effecteur terminal est ensuite transformé des coordonnées (0, -0,2, 0) à (0, -0,394, 0,694) par rapport au repère de base.

Chapitre 4. Application des quaternions à Universal Robot UR3

La fonction record pose est définie pour utiliser la représentation axe-angle. À des fins de démonstration, la dérivation de la cinématique directe à partir d'une matrice de transformation peut également être effectuée en appelant la fonction de cinématique directe en définissant les angles d'articulation comme paramètres et TM comme mode de transformation. La mise en œuvre est illustrée ci-dessous :

```
>>> get_forward_kinematics("TM",rob.getj())

Current Joint Angles: [0, -1.5707963267948966, 0, -1.5707963267948966, 0, 0]

Deriving Forward Kinematics from Transformation Matrix...
Computing Transformation Matrix by Denavit-Hartenberg Parameters...

Transformation Matrix for Frame 0 to 1
[[ 1.  -0.  0.  0. ]
 [ 0.  1.  -0.  -0. ]
 [ 0.  0.  1.  0.1519]
 [ 0.  0.  0.  1.  ]]
Dual Quaternion:([ 1.000 +0.000i +0.000j +0.000k ],< 0.000 +0.000i -0.000j +0.152k >)

Transformation Matrix for Frame 1 to 2
[[ 6.12323400e-17  1.00000000e+00  0.00000000e+00  0.00000000e+00]
 [-6.12323400e-17  3.74939946e-33 -1.00000000e+00 -0.00000000e+00]
 [-1.00000000e+00  6.12323400e-17  6.12323400e-17  0.00000000e+00]
 [ 0.00000000e+00  0.00000000e+00  0.00000000e+00  1.00000000e+00]]
Dual Quaternion:([ 0.500 +0.500i +0.500j -0.500k ],< 0.000 +0.000i -0.000j +0.000k >)

Transformation Matrix for Frame 2 to 3
[[ 1.  -0.  0.  -0.24365]
 [ 0.  1.  -0.  -0. ]
 [ 0.  0.  1.  0. ]
 [ 0.  0.  0.  1.  ]]
Dual Quaternion:([ 1.000 +0.000i +0.000j +0.000k ],< 0.000 -0.244i -0.000j +0.000k >)

Transformation Matrix for Frame 3 to 4
[[ 6.123234e-17  1.000000e+00  0.000000e+00 -2.132500e-01]
 [-1.000000e+00  6.123234e-17 -0.000000e+00 -0.000000e+00]
 [-0.000000e+00  0.000000e+00  1.000000e+00  1.123500e-01]
 [ 0.000000e+00  0.000000e+00  0.000000e+00  1.000000e+00]]
Dual Quaternion:([ 0.707 +0.000i +0.000j -0.707k ],< 0.000 -0.213i -0.000j +0.112k >)

Transformation Matrix for Frame 4 to 5
[[ 1.00000000e+00 -0.00000000e+00  0.00000000e+00  0.00000000e+00]
 [ 0.00000000e+00  6.12323400e-17 -1.00000000e+00 -8.53500000e-02]
 [ 0.00000000e+00  1.00000000e+00  6.12323400e-17  5.22618022e-18]
 [ 0.00000000e+00  0.00000000e+00  0.00000000e+00  1.00000000e+00]]
Dual Quaternion:([ 0.707 +0.707i +0.000j +0.000k ],< 0.000 +0.000i -0.085j +0.000k >)

Transformation Matrix for Frame 5 to 6
[[ 1.00000000e+00 -0.00000000e+00  0.00000000e+00  0.00000000e+00]
 [ 0.00000000e+00  6.12323400e-17  1.00000000e+00  8.19000000e-02]
 [-0.00000000e+00 -1.00000000e+00  6.12323400e-17  5.01492864e-18]
 [ 0.00000000e+00  0.00000000e+00  0.00000000e+00  1.00000000e+00]]
Dual Quaternion:([ 0.707 -0.707i +0.000j +0.000k ],< 0.000 +0.000i +0.082j +0.000k >)

Forward Kinematics (by Dual Quaternion):
([ 0.000 +0.000i +0.707j -0.707k ],< 0.000 +0.000i -0.194j +0.694k >)

Initial TCP Coordinates (at zero pose): [0.  0.  0.2]

After Rotation: [ 0.  -0.2  0. ]
After Translation: [ 0.  -0.39425  0.69415]

Transformed TCP Coordinates:
[ 0.  -0.394  0.694]
```

Le quaternion double représentant la transformation finale dans ces deux approches est clairement différent mais les coordonnées transformées sont les mêmes. Notez que nous utilisons également différentes coordonnées TCP initiales.

4.5. Vérification des résultats

La fonction `record_pose` stocke les poses dans un fichier et la fonction `run_poses` récupère les coefficients des quaternions doubles stockés dans le fichier et génère les angles correspondant à ces transformations. Ceci est démontré ci-dessous :

```
Python 3.7.2 Shell
File Edit Shell Debug Options Window Help
>>> run_poses()

Desired Pose: -0.000 +0.000i +1.000j -0.000k 0.000 -0.000i -0.194j +0.694k

Set of Possible Joint Angles:
[[6.283185307179586, -1.570796340704604, 2.9802322387695312e-08, 4.7123889644920745,
0.0, 0], [6.283185307179586, -1.5707963128851894, -2.9802322387695312e-08, 4.712388996
277305, 0.0, 0], [0.0, -1.5707963407046037, 2.9802322387695312e-08, -1.570796342687512
, 0.0, 0], [0.0, -1.5707963128851894, -2.9802322387695312e-08, -1.5707963109022816, 0.
0, 0]]

Current Joint Angles: [6.283185307179586, -1.570796340704604, 2.9802322387695312e-08,
4.7123889644920745, 0.0, 0]

Deriving Forward Kinematics from Angle-Axis Representation...
Dual Quaternion: ([ -1.000 +0.000i -0.000j +0.000k ], < 0.000 +0.000i +0.000j +0.152k >)
Dual Quaternion: ([ 0.707 -0.000i +0.707j -0.000k ], < 0.000 +0.000i +0.000j +0.244k >)
Dual Quaternion: ([ 1.000 +0.000i -0.000j +0.000k ], < 0.000 -0.213i -0.112j -0.000k >)
Dual Quaternion: ([ -0.707 +0.000i -0.707j +0.000k ], < 0.000 -0.085i +0.000j +0.000k >)
Dual Quaternion: ([ 1.000 +0.000i -0.000j -0.000k ], < 0.000 -0.000i -0.082j +0.000k >)
Dual Quaternion: ([ 1.000 +0.000i -0.000j +0.000k ], < 0.000 +0.000i +0.000j +0.000k >)

Forward Kinematics (by Dual Quaternion):
([ -0.000 +0.000i +1.000j -0.000k ], < 0.000 +0.000i -0.194j +0.694k >)

Initial TCP Coordinates (at zero pose): [ 0. -0.2 0. ]

After Rotation: [-4.8985872e-17 -2.0000000e-01 2.4492936e-17]
After Translation: [-3.60044906e-17 -3.94250000e-01 6.94150000e-01]

Transformed TCP Coordinates:
[-0. -0.394 0.694]

Comparing Dual Quaternions:
Desired Pose:
([ -0.000 +0.000i +1.000j -0.000k ], < 0.000 -0.000i -0.194j +0.694k >)
Current Pose:
([ -0.000 +0.000i +1.000j -0.000k ], < 0.000 +0.000i -0.194j +0.694k >)
PASSED criteria!
```

Il y avait 4 combinaisons d'angles possibles. Ceci est en outre vérifié en vérifiant si sa transformation correspondante (représentée par le quaternion double) est approximativement égale à notre transformation souhaitée, comme également vu ci-dessus.

Les résultats finaux à la fin sont donnés par,

```
Final Joint Angles: [[6.283185307179586, -1.570796340704604, 2.9802322387695312e-08, 4
.7123889644920745, 0.0, 0], [6.283185307179586, -1.5707963128851894, -2.98023223876953
12e-08, 4.712388996277305, 0.0, 0], [0.0, -1.5707963407046037, 2.9802322387695312e-08,
-1.570796342687512, 0.0, 0], [0.0, -1.5707963128851894, -2.9802322387695312e-08, -1.57
07963109022816, 0.0, 0]]

There are 4 possible pose(s)!
>>>
```

Par conséquent, toutes les combinaisons étaient valides et toutes transformeront le point central de l'outil de notre robot dans la pose souhaitée (mêmes coordonnées TCP transformées) qui est la solution de la cinématique inverse.

Chapitre 5

Conclusions et Recommandations

5.1 Conclusion

La thèse a pu démontrer avec succès l'utilisation de l'algèbre des quaternions dans la représentation de cinématiques pour des applications robotiques notamment avec le Robot Universel 6-DOF UR3. Sa cinématique directe et inverse ont été dérivés en utilisant les principes des quaternions. Ceux-ci sont contenus dans Section 4.2 Cinématique directe par quaternions doubles et Section 4.3 Inverse Cinématique par doubles quaternions. Une application de programmation pratique a été également développé montré dans la section 4.4.2 Application de programmation Python pour confirmer l'exactitude de la cinématique dérivée.

En dehors de cela, sur la base de notre analyse des résultats, nous pouvons également conclure les déclarations suivantes :

1. Les quaternions doubles présentent une manière plus compacte de représenter les transformations tridimensionnelles. Ainsi, offre également un avantage considérable en termes d'efficacité de stockage.
 - (a) Cela est dû au fait que les quaternions ne nécessitent que quatre éléments pour représenter une rotation par rapport à la rotation la plus couramment utilisée matrices qui implique 9 éléments.
 - (b) Pour la transformation en trois dimensions qui implique rotation et translation. Une matrice de transformation homogène est constituée de 16 éléments dont quatre éléments sont triviaux tandis qu'un quaternion n'a besoin que de 7 éléments pour représenter la même transformation.

Chapitre 5. Conclusions et recommandations

2. Effectuer des transformations multiples par des quaternions doubles présente également un avantage en termes de robustesse de calcul et de précision numérique, en particulier pour traiter la cinématique d'un robot.
 - (a) Combiner des transformations à l'aide de matrices de transformation homogènes implique de multiplier des matrices 4×4 , ce qui implique plusieurs opérations arithmétiques et trigonométriques qui peuvent grandement diminuer la précision numérique de la solution.
 - (b) L'exécution de la multiplication de quaternions est comparable à la multiplication polynomiale tout en incorporant également les lois de multiplication pour les unités de quaternions. Cela le rend plus efficace, et aussi plus précis numériquement puisqu'il implique moins d'opérations arithmétiques et trigonométriques.
3. Les quaternions offrent également un grand avantage en termes de normalisation et facilité de dérivation.
 - (a) Les quaternions sont plus faciles à normaliser que les matrices. Une matrice de rotation doit être orthogonale et doit avoir un déterminant de 1.
Pour les quaternions, il doit s'agir d'un quaternion unitaire, ce qui signifie qu'il doit avoir une norme d'unité. Cela peut être simplement fait en divisant les quatre éléments des quaternions par sa magnitude.
 - (b) Les quaternions doubles sont facilement dérivés en utilisant la représentation axe-angle dans laquelle nous avons seulement besoin de déterminer la relation entre l'axe de rotation actuel et le cadre de coordonnées de base.
4. Les doubles quaternions ne sont pas sensibles au verrouillage du cardan.
 - (a) Bien qu'il ne soit pas abordé dans cette étude, le verrouillage du cardan est un problème sérieux dans le traitement des rotations dans l'espace tridimensionnel. Imaginez trois cardans qui représentent des rotations autour de chacun des trois axes ; ainsi, représente les trois degrés de liberté. Ceux-ci sont définis comme le roulis, le tangage et le lacet. Cela se produit lorsque les cardans extérieurs et intérieurs s'alignent et perdent un degré de liberté. Cela est dû au fait que la rotation qui peut être effectuée par l'un de ces deux cardans est désormais considérée comme la même. Euler Angles est connu pour être sensible à cet événement rare.
 - (b) Les quaternions comme les matrices de rotation sont connus pour ne pas être sensibles au blocage de cardan. Cela est dû au fait qu'ils représentent l'orientation comme une valeur plutôt que comme trois valeurs distinctes et liées.

5. La facilité de représentation de la cinématique du robot par des quaternions doubles dépend du type de cinématique dérivée et de la structure cinématique du robot.

(a) Pour la cinématique directe, il est conseillé de dériver les rotations par une approche axe-angle plutôt que de partir d'une matrice de rotation, puis de la convertir en quaternions unitaires. De cette façon, il y aura clairement moins d'opérations à effectuer par rapport à la transformation homogène traditionnelle.

(b) Pour la cinématique inverse, la difficulté de dériver des formules pour les angles est toujours préservée dans les quaternions doubles. Cela dépend généralement de la structure cinématique du robot, comme dans notre cas. Il s'agit de la relation entre la transformation qui peut être effectuée par chaque joint et la transformation finale. Dans notre cas, nous avons dû diviser les transformations en trois. Cependant dans certains cas, il peut arriver que certaines formules ne puissent déjà être dérivées que par les équations générées dans la transformation finale.

(c) Afin de toujours appliquer le concept de quaternions dans la cinématique inverse, nous avons également dû analyser et calculer plusieurs équations pouvant représenter les transformations effectuées par chaque articulation. Cela le rend plus fastidieux par rapport à la convention normale qui est généralement effectuée en utilisant l'approche de la solution géométrique.

Espérons que ces avantages seront jugés suffisants pour que les quaternions doubles soient utilisés plus largement non seulement dans le domaine de la robotique et de l'infographie, mais également dans d'autres domaines qui impliquent de travailler dans un espace tridimensionnel.

5.2 Recommendation

Pour les recherches futures, il est recommandé de trouver s'il existe une manière meilleure et plus efficace de représenter la cinématique du robot, en particulier pour la cinématique inverse. Pour l'Universal Robot UR3, la solution cinématique inverse peut être davantage filtrée en ne considérant qu'une certaine plage de mouvements.

Certaines conditions peuvent également être appliquées pour déterminer la solution optimale qui donnera la pose désirée. Comme cette recherche s'est concentrée uniquement sur l'application des quaternions dans la cinématique des robots, elle peut être étendue à son application dans le contrôle des robots en général. Cela peut impliquer une étape supplémentaire consistant à effectuer une interpolation quaternionique. Cela contribue à trouver la trajectoire optimale pour la trajectoire du robot. Ce sera bon pour les recherches futures car c'est aussi l'une des applications utiles et bien connues des quaternions.

Annexes

Annexe A

Robot Universel UR3

Spécifications techniques

UR3

Performance	
Répétabilité	±0,1 mm / ±0,0039 pouces (4 mils)
Plage de température ambiante	0-50° *
Consommation d'énergie	Min 90W, Typique 125W, Max 250W
Opération collaborative	15 fonctions de sécurité réglables avancées. Fonction de sécurité approuvée par TÜV NORD Testé conformément à : EN ISO 13849:2008 PL d

spécification	
Charge utile	3 kg / 6,6 livres
Atteindre	500 mm / 19,7 pouces
Degrés de liberté	6 articulations tournantes
La programmation	Interface utilisateur graphique Polyscope sur écran tactile 12 pouces avec fixation

Mouvement		
Bras de robot à mouvement d'axe	Plage de travail	Vitesse maximum
Base	± 360°	± 180°/Sec.
Épaule	± 360°	± 180°/Sec.
Coude	± 360°	± 180°/Sec.
Poignet 1	± 360°	± 360°/Sec.
Poignet 2	± 360°	± 360°/Sec.
Poignet 3	Infini	±360°/Sec.
Outil typique		1 m/s. / 39,4 po/s.

Caractéristiques

Classement IP	IP64	
Salle blanche de classe ISO	5	
Bruit	70dB	
Montage de robots	Modulaire Select	
Ports d'E/S	Entrée numérique	2
	Numérique	2
	Entrée analogique	2
	Sortie analogique	0
Alimentation E/S dans l'outil	12 V/24 V 600 mA dans l'outil	

Physique	
Empreinte	Ø 128mm
Matériaux	Aluminium, plastiques PP
Type de connecteur d'outil	M8
Bras de robot de longueur de câble	6 mètres / 236 pouces
Poids avec câble	11 kg / 24,3 lb

* Le robot peut fonctionner dans une plage de température de 0 à 50°C. À vitesse de joint continue élevée, la température ambiante est réduite.

BOÎTIER DE COMMANDE

Caractéristiques		
Classement IP	IP20	
Salle blanche de classe ISO	6	
Bruit	<65dB(A)	
Ports d'E/S	Entrée numérique	16
	Numérique	16
	Entrée analogique	2
	Sortie analogique	2
Alimentation E/S	24V 2A	
Communication	TCP/IP 100 Mbits, Modbus TCP, Profinet,EthernetIP	
Source d'énergie	100-240 VCA, 50-60 Hz	
Plage de température ambiante	0-50°	
Physique		
Taille du boîtier de commande (LxHxP)	475 mm x 423 mm x 268 mm / 18,7 x 16,7 x 10,6 pouces	
Poids	15 kg / 33,1 lb	
Matériaux	Acier	

PENDENTIF D'ENSEIGNEMENT

Caractéristiques	
Classement IP	IP20
Physique	
Matériaux	Aluminium, polypropylène
Poids	1,5 kg / 3,3 livres
Longueur de câble	4,5 m / 177 po



Appendice B

Produits de transformation pour Base au poignet 3

Cette partie présente les produits de transformation pour la cinématique du robot UR3 structure du cadre 1 (base) au cadre 6 (poignet 3).

Nous définissons le produit de quaternion dual M_i comme :

$$M_i = Q_i \quad M_{i+1} \text{ où } 1 \leq i \leq 6$$

$$M_6 = Q_6$$

$$M_5 = Q_5 \quad M_6 = Q_5 \quad Q_6$$

$$M_4 = Q_4 \quad M_5 = Q_4 \quad Q_5 \quad Q_5$$

$$M_3 = Q_3 \quad M_4 = Q_3 \quad Q_4 \quad Q_5 \quad Q_6$$

$$M_2 = Q_2 \quad M_3 = Q_2 \quad Q_3 \quad Q_4 \quad Q_5 \quad Q_6$$

$$M_1 = Q_1 \quad M_2 = Q_1 \quad Q_2 \quad Q_3 \quad Q_4 \quad Q_5 \quad Q_6$$

Nous définissons le produit quaternionique N_i comme suit :

$$N_{i+1} = Q^{-1}_i \quad N_i \text{ où } 1 \leq i \leq 6$$

$$N_1 = QFK = ([w, < a, b, c >], < x, y, z >)$$

$$N_2 = Q^{-1}_1 \quad N_1 = Q^{-1}_1 \quad QFK$$

$$N_3 = Q^{-1}_2 \quad N_2 = Q^{-1}_2 \quad Q^{-1}_1 \quad QFK$$

$$N_4 = Q^{-1}_3 \quad N_3 = Q^{-1}_3 \quad Q^{-1}_2 \quad Q^{-1}_1 \quad QFK$$

$$N_5 = Q^{-1}_4 \quad N_4 = Q^{-1}_4 \quad Q^{-1}_3 \quad Q^{-1}_2 \quad Q^{-1}_1 \quad QFK$$

$$N_6 = Q^{-1}_5 \quad N_5 = Q^{-1}_5 \quad Q^{-1}_4 \quad Q^{-1}_3 \quad Q^{-1}_2 \quad Q^{-1}_1 \quad QFK$$

Annexe B. Produits de transformation de la base au poignet 3

Produits de transformation de la base au poignet 3 ($M6 = N6$)

$$M6 = ([M61 + M62i + M63j + M64k], < M65i + M66j + M67k >)$$

$$N6 = ([N61 + N62i + N63j + N64k], < N65i + N66j + N67k >)$$

$$\begin{array}{rclcl}
 & & \underline{M61 = N61} & & \\
 \text{parce que } \frac{\theta_6}{2} & = w \cos & \frac{\theta_2 + \theta_3 + \theta_4}{2} & \text{parce que } \frac{\theta_1 - \theta_5}{2} & (B.1) \\
 & + \text{un péché} & \frac{\theta_2 + \theta_3 + \theta_4}{2} & \text{péché} & \frac{\theta_1 + \theta_5}{2} \\
 & -b \text{ péché} & \frac{\theta_2 + \theta_3 + \theta_4}{2} & \text{parce que } \frac{\theta_1 + \theta_5}{2} & \\
 & +c \cos & \frac{\theta_2 + \theta_3 + \theta_4}{2} & \text{péché} & \frac{\theta_1 - \theta_5}{2}
 \end{array}$$

$$\begin{array}{rclcl}
 & & \underline{M62 = N62} & & \\
 0 = -w \sin & & \frac{\theta_2 + \theta_3 + \theta_4}{2} & \text{péché} & \frac{\theta_1 + \theta_5}{2} & (B.2) \\
 & + \text{un cos} & \frac{\theta_2 + \theta_3 + \theta_4}{2} & \text{parce que } \frac{\theta_1 - \theta_5}{2} & \\
 & +b \cos & \frac{\theta_2 + \theta_3 + \theta_4}{2} & \text{péché} & \frac{\theta_1 - \theta_5}{2} \\
 & +c \text{ péché} & \frac{\theta_2 + \theta_3 + \theta_4}{2} & \text{parce que } \frac{\theta_1 + \theta_5}{2}
 \end{array}$$

$$\begin{array}{rclcl}
 & & \underline{M63 = N63} & & \\
 - \text{péché} \frac{\theta_6}{2} & = w \text{ péché} & \frac{\theta_2 + \theta_3 + \theta_4}{2} & \text{parce que } \frac{\theta_1 + \theta_5}{2} & (B.3) \\
 & -a \cos & \frac{\theta_2 + \theta_3 + \theta_4}{2} & \text{péché} & \frac{\theta_1 - \theta_5}{2} \\
 & +b \cos & \frac{\theta_2 + \theta_3 + \theta_4}{2} & \text{parce que } \frac{\theta_1 - \theta_5}{2} \\
 & +c \text{ péché} & \frac{\theta_2 + \theta_3 + \theta_4}{2} & \text{péché} & \frac{\theta_1 + \theta_5}{2}
 \end{array}$$

$$\underline{M64 = N64}$$

$$\begin{aligned}
 0 = -w \cos \frac{\theta_2 + \theta_3 + \theta_4}{2} & \quad \text{péché} \quad \frac{\theta_1 - \theta_5}{2} & (B.4) \\
 -un \text{ péché} \frac{\theta_2 + \theta_3 + \theta_4}{2} & \quad \text{parce que} \quad \frac{\theta_1 + \theta_5}{2} \\
 -b \text{ péché} \frac{\theta_2 + \theta_3 + \theta_4}{2} & \quad \text{péché} \quad \frac{\theta_1 + \theta_5}{2} \\
 +c \cos \frac{\theta_2 + \theta_3 + \theta_4}{2} & \quad \text{parce que} \quad \frac{\theta_1 - \theta_5}{2}
 \end{aligned}$$

$$\underline{M65 = N65}$$

$$\begin{aligned}
 0 = & -d_1 \cos \theta_5 \sin (\theta_2 + \theta_3 + \theta_4) - d_4 \sin \theta_5 - a_2 \cos (\theta_3 + \theta_4) \cos \theta_5 \quad (B.5) \\
 & - a_3 \cos \theta_4 \cos \theta_5 + x [\cos \theta_1 \cos (\theta_2 + \theta_3 + \theta_4) \cos \theta_5 + \sin \theta_1 \sin \theta_5] \\
 & + y [\cos (\theta_2 + \theta_3 + \theta_4) \cos \theta_5 \sin \theta_1 - \cos \theta_1 \sin \theta_5] \\
 & + z \cos \theta_5 \sin (\theta_2 + \theta_3 + \theta_4)
 \end{aligned}$$

$$\underline{M66 = N66}$$

$$\begin{aligned}
 0 = & -d_1 \sin (\theta_2 + \theta_3 + \theta_4) \sin \theta_5 + d_4 \cos \theta_5 - a_2 \cos (\theta_3 + \theta_4) \sin \theta_5 \quad (B.6) \\
 & - a_3 \cos \theta_4 \sin \theta_5 + x [\cos \theta_1 \cos (\theta_2 + \theta_3 + \theta_4) \sin \theta_5 - \cos \theta_5 \sin \theta_1] \\
 & + y [\cos (\theta_2 + \theta_3 + \theta_4) \sin \theta_1 \sin \theta_5 + \cos \theta_1 \cos \theta_5] \\
 & + z \sin (\theta_2 + \theta_3 + \theta_4) \sin \theta_5 + d_6
 \end{aligned}$$

$$\underline{M67 = N67}$$

$$\begin{aligned}
 0 = & -d_1 \cos (\theta_2 + \theta_3 + \theta_4) - x \cos \theta_1 \sin (\theta_2 + \theta_3 + \theta_4) - y \sin & (B.7) \\
 & \theta_1 \sin (\theta_2 + \theta_3 + \theta_4) + z \cos (\theta_2 + \theta_3 + \theta_4) \\
 & + a_2 \sin (\theta_3 + \theta_4) + a_3 \sin \theta_4 + d_5
 \end{aligned}$$

Annexe B. Produits de transformation de la base au poignet 3

Produits de transformation de la base au poignet 3 ($M5 = N5$)

$$M5 = ([M51 + M52i + M53j + M54k], < M55i + M56j + M57k >)$$

$$N5 = ([N51 + N52i + N53j + N54k], < N55i + N56j + N57k >)$$

$$\underline{M51 = N51}$$

$$\begin{aligned} \text{parce que } \frac{\theta 5}{2} \quad \text{parce que } \frac{\theta 6}{2} &= w \cos \frac{\theta 1}{2} \quad \text{parce que } \frac{\theta 2 + \theta 3 + \theta 4}{2} \\ &+ \text{un péché } \frac{\theta 1}{2} \quad \text{péché } \frac{\theta 2 + \theta 3 + \theta 4}{2} \\ &- b \cos \frac{\theta 1}{2} \quad \text{péché } \frac{\theta 2 + \theta 3 + \theta 4}{2} \\ &+ c \text{ péché } \frac{\theta 1}{2} \quad \text{parce que } \frac{\theta 2 + \theta 3 + \theta 4}{2} \end{aligned} \quad (B.8)$$

$$\underline{M52 = N52}$$

$$\begin{aligned} - \text{péché } \frac{\theta 5}{2} \quad \text{péché } \frac{\theta 6}{2} &= -w \text{ péché } \frac{\theta 1}{2} \quad \text{péché } \frac{\theta 2 + \theta 3 + \theta 4}{2} \\ &+ \text{un cos } \frac{\theta 1}{2} \quad \text{parce que } \frac{\theta 2 + \theta 3 + \theta 4}{2} \\ &+ b \text{ péché } \frac{\theta 1}{2} \quad \text{parce que } \frac{\theta 2 + \theta 3 + \theta 4}{2} \\ &+ c \cos \frac{\theta 1}{2} \quad \text{péché } \frac{\theta 2 + \theta 3 + \theta 4}{2} \end{aligned} \quad (B.9)$$

$$\underline{M53 = N53}$$

$$\begin{aligned} - \cos \frac{\theta 5}{2} \quad \text{péché } \frac{\theta 6}{2} &= w \cos \frac{\theta 1}{2} \quad \text{péché } \frac{\theta 2 + \theta 3 + \theta 4}{2} \\ &- \text{un péché } \frac{\theta 1}{2} \quad \text{parce que } \frac{\theta 2 + \theta 3 + \theta 4}{2} \\ &+ b \cos \frac{\theta 1}{2} \quad \text{parce que } \frac{\theta 2 + \theta 3 + \theta 4}{2} \\ &+ c \text{ péché } \frac{\theta 1}{2} \quad \text{péché } \frac{\theta 2 + \theta 3 + \theta 4}{2} \end{aligned} \quad (B.10)$$

$$M54 = N54$$

$$- \text{péché} \frac{\theta_5}{2} \text{ parce que } \frac{\theta_6}{2} = -w \text{ péché} \frac{\theta_1}{2} \text{ parce que } \frac{\theta_2 + \theta_3 + \theta_4}{2} \quad (\text{B.11})$$

$$-a \cos \frac{\theta_1}{2} \text{ péché} \frac{\theta_2 + \theta_3 + \theta_4}{2}$$

$$-b \text{ péché} \frac{\theta_1}{2} \text{ péché} \frac{\theta_2 + \theta_3 + \theta_4}{2}$$

$$+c \cos \frac{\theta_1}{2} \text{ parce que } \frac{\theta_2 + \theta_3 + \theta_4}{2}$$

$$M55 = N55$$

$$-d_6 \sin \theta_5 = -d_1 \sin (\theta_2 + \theta_3 + \theta_4) - a_2 \cos (\theta_3 + \theta_4) - a_3 \cos \theta_4 \quad (\text{B.12})$$

$$+ x \cos \theta_1 \cos (\theta_2 + \theta_3 + \theta_4) + y \sin \theta_1 \cos (\theta_2 + \theta_3 + \theta_4)$$

$$+ z \sin (\theta_2 + \theta_3 + \theta_4)$$

$$M56 = N56$$

$$-d_6 \cos \theta_5 = y \cos \theta_1 - x \sin \theta_1 + d_4 \quad (\text{B.13})$$

$$M57 = N57$$

$$0 = -d_1 \cos (\theta_2 + \theta_3 + \theta_4) + d_5 + a_2 \sin (\theta_3 + \theta_4) + a_3 \sin \theta_4 - x \cos \theta_1 \sin (\theta_2 + \theta_3 +$$

$$\theta_4) - y \sin \theta_1 \sin (\theta_2 + \theta_3 + \theta_4)$$

$$+ z \cos (\theta_2 + \theta_3 + \theta_4)$$

Annexe B. Produits de transformation de la base au poignet 3

Produits de transformation de la base au poignet 3 ($M4 = N4$)

$$M4 = ([M41 + M42i + M43j + M44k], < M45i + M46j + M47k >)$$

$$N4 = ([N41 + N42i + N43j + N44k], < N45i + N46j + N47k >)$$

$$\underline{M41 = N41}$$

$$\begin{aligned} \text{parce que } \frac{\theta 5}{2} \quad \text{parce que } \frac{\theta 4 + \theta 6}{2} &= w \cos \frac{\theta 1}{2} \quad \text{parce que } \frac{\theta 2 + \theta 3}{2} & (B.15) \\ + \text{un péché } \frac{\theta 1}{2} \quad \text{péché } \frac{\theta 2 + \theta 3}{2} \\ -b \cos \frac{\theta 1}{2} \quad \text{péché } \frac{\theta 2 + \theta 3}{2} \\ +c \text{ péché } \frac{\theta 1}{2} \quad \text{parce que } \frac{\theta 2 + \theta 3}{2} \end{aligned}$$

$$\underline{M42 = N42}$$

$$\begin{aligned} \text{péché } \frac{\theta 5}{2} \quad \text{péché } \frac{\theta 4 - \theta 6}{2} &= -w \text{ péché } \frac{\theta 1}{2} \quad \text{péché } \frac{\theta 2 + \theta 3}{2} & (B.16) \\ + \text{un cos } \frac{\theta 1}{2} \quad \text{parce que } \frac{\theta 2 + \theta 3}{2} \\ +b \text{ péché } \frac{\theta 1}{2} \quad \text{parce que } \frac{\theta 2 + \theta 3}{2} \\ +c \cos \frac{\theta 1}{2} \quad \text{péché } \frac{\theta 2 + \theta 3}{2} \end{aligned}$$

$$\underline{M43 = N43}$$

$$\begin{aligned} - \cos \frac{\theta 5}{2} \quad \text{péché } \frac{\theta 4 + \theta 6}{2} &= w \cos \frac{\theta 1}{2} \quad \text{péché } \frac{\theta 2 + \theta 3}{2} & (B.17) \\ - \text{un péché } \frac{\theta 1}{2} \quad \text{parce que } \frac{\theta 2 + \theta 3}{2} \\ +b \cos \frac{\theta 1}{2} \quad \text{parce que } \frac{\theta 2 + \theta 3}{2} \\ +c \text{ péché } \frac{\theta 1}{2} \quad \text{péché } \frac{\theta 2 + \theta 3}{2} \end{aligned}$$

$$\underline{M44 = N44}$$

$$\begin{aligned}
 -\text{péché} \frac{\theta_5}{2} \quad \text{parce que} \quad \frac{\theta_4 - \theta_6}{2} &= -w \text{ péché} \frac{\theta_1}{2} \quad \text{parce que} \quad \frac{\theta_2 + \theta_3}{2} & (B.18) \\
 -a \cos \frac{\theta_1}{2} \quad \text{péché} \quad \frac{\theta_2 + \theta_3}{2} \\
 -b \text{ péché} \frac{\theta_1}{2} \quad \text{péché} \quad \frac{\theta_2 + \theta_3}{2} \\
 +c \cos \frac{\theta_1}{2} \quad \text{parce que} \quad \frac{\theta_2 + \theta_3}{2}
 \end{aligned}$$

$$\underline{M45 = N45}$$

$$-d_6 \cos \theta_4 \sin \theta_5 + d_5 \sin \theta_4 = -d_1 \sin (\theta_2 + \theta_3) - a_2 \cos \theta_3 - a_3 + x \cos \theta_1 \cos (\theta_2 + \theta_3) \quad (B.19)$$

$$+ y \sin \theta_1 \cos (\theta_2 + \theta_3)$$

$$+ z \sin (\theta_2 + \theta_3)$$

$$\underline{M46 = N46}$$

$$-d_6 \cos \theta_5 = y \cos \theta_1 - x \sin \theta_1 + d_4 \quad (B.20)$$

$$\underline{M47 = N47}$$

$$-d_6 \sin \theta_4 \sin \theta_5 - d_5 \cos \theta_4 = -d_1 \cos (\theta_2 + \theta_3) + a_2 \sin \theta_3 - x \cos \theta_1 \sin (\theta_2 + \theta_3) \quad (B.21)$$

$$- y \sin \theta_1 \sin (\theta_2 + \theta_3)$$

$$+ z \cos (\theta_2 + \theta_3)$$

Annexe B. Produits de transformation de la base au poignet 3

Produits de transformation de la base au poignet 3 ($M3 = N3$)

$$M3 = ([M31 + M32i + M33j + M34k], < M35i + M36j + M37k >)$$

$$N3 = ([N31 + N32i + N33j + N34k], < N35i + N36j + N37k >)$$

$$\underline{M31 = N31}$$

$$\begin{aligned} \text{parce que } \frac{\theta 5}{2} \quad \text{parce que } \frac{\theta 3 + \theta 4 + \theta 6}{2} &= w \cos \frac{\theta 1}{2} \quad \text{parce que } \frac{\theta 2}{2} & (B.22) \\ &+ \text{un péché } \frac{\theta 1}{2} \quad \text{péché } \frac{\theta 2}{2} \\ &- b \cos \frac{\theta 1}{2} \quad \text{péché } \frac{\theta 2}{2} \\ &+ c \text{ péché } \frac{\theta 1}{2} \quad \text{parce que } \frac{\theta 2}{2} \end{aligned}$$

$$\underline{M32 = N32}$$

$$\begin{aligned} \text{péché } \frac{\theta 5}{2} \quad \text{péché } \frac{\theta 3 + \theta 4 - \theta 6}{2} &= -w \text{ péché } \frac{\theta 1}{2} \quad \text{péché } \frac{\theta 2}{2} & (B.23) \\ &+ \text{un cos } \frac{\theta 1}{2} \quad \text{parce que } \frac{\theta 2}{2} \\ &+ b \text{ péché } \frac{\theta 1}{2} \quad \text{parce que } \frac{\theta 2}{2} \\ &+ c \cos \frac{\theta 1}{2} \quad \text{péché } \frac{\theta 2}{2} \end{aligned}$$

$$\underline{M33 = N33}$$

$$\begin{aligned} - \cos \frac{\theta 5}{2} \quad \text{péché } \frac{\theta 3 + \theta 4 + \theta 6}{2} &= w \cos \frac{\theta 1}{2} \quad \text{péché } \frac{\theta 2}{2} & (B.24) \\ &- \text{un péché } \frac{\theta 1}{2} \quad \text{parce que } \frac{\theta 2}{2} \\ &+ b \cos \frac{\theta 1}{2} \quad \text{parce que } \frac{\theta 2}{2} \\ &+ c \text{ péché } \frac{\theta 1}{2} \quad \text{péché } \frac{\theta 2}{2} \end{aligned}$$

$$\underline{M34 = N34}$$

$$\begin{aligned}
 -\text{péché} \frac{\theta_5}{2} \quad \text{parce que} \quad \frac{\theta_3 + \theta_4 - \theta_6}{2} &= -w \text{ péché} \frac{\theta_1}{2} \quad \text{parce que} \quad \frac{\theta_2}{2} & (B.25) \\
 -a \cos \frac{\theta_1}{2} &\quad \text{péché} \quad \frac{\theta_2}{2} \\
 -b \text{ péché} \frac{\theta_1}{2} &\quad \text{péché} \quad \frac{\theta_2}{2} \\
 +c \cos \frac{\theta_1}{2} &\quad \text{parce que} \quad \frac{\theta_2}{2}
 \end{aligned}$$

$$\underline{M35 = N35}$$

$$\begin{aligned}
 -d_6 \cos(\theta_3 + \theta_4) \sin \theta_5 + d_5 \sin(\theta_3 + \theta_4) + a_3 \cos \theta_3 &= -d_1 \sin \theta_2 - a_2 + x & (B.26) \\
 \cos \theta_1 \cos \theta_2 \\
 + y \sin \theta_1 \cos \theta_2 + z \sin \theta_2
 \end{aligned}$$

$$\underline{M36 = N36}$$

$$-d_6 \cos \theta_5 - d_4 = y \cos \theta_1 - x \sin \theta_1 \quad (B.27)$$

$$\underline{M37 = N37}$$

$$\begin{aligned}
 -d_6 \sin(\theta_3 + \theta_4) \sin \theta_5 - d_5 \cos(\theta_3 + \theta_4) + a_3 \sin \theta_3 &= & (B.28) \\
 -d_1 \cos \theta_2 - x \cos \theta_1 \sin \theta_2 \\
 -y \sin \theta_1 \sin \theta_2 + z \cos \theta_2
 \end{aligned}$$

Annexe B. Produits de transformation de la base au poignet 3

Produits de transformation de la base au poignet 3 (M2 = N2)

$$M2 = ([M21 + M22i + M23j + M24k], < M25i + M26j + M27k >)$$

$$N2 = ([N21 + N22i + N23j + N24k], < N25i + N26j + N27k >)$$

$$M21 = N21$$

$$\text{parce que } \frac{\theta_5}{2} \quad \text{parce que } \frac{\theta_2 + \theta_3 + \theta_4 + \theta_6}{2} = w \cos \frac{\theta_1}{2} + c \text{ péché } \frac{\theta_1}{2} \quad (B.29)$$

$$M22 = N22$$

$$\text{péché } \frac{\theta_5}{2} \quad \text{péché } \frac{\theta_2 + \theta_3 + \theta_4 - \theta_6}{2} = \text{un } \cos \frac{\theta_1}{2} + b \text{ péché } \frac{\theta_1}{2} \quad (B.30)$$

$$M23 = N23$$

$$- \cos \frac{\theta_5}{2} \quad \text{péché } \frac{\theta_2 + \theta_3 + \theta_4 + \theta_6}{2} = -\text{un péché } \frac{\theta_1}{2} + b \cos \frac{\theta_1}{2} \quad (B.31)$$

$$M24 = N24$$

$$- \text{péché } \frac{\theta_5}{2} \quad \text{parce que } \frac{\theta_2 + \theta_3 + \theta_4 - \theta_6}{2} = -w \text{ péché } \frac{\theta_1}{2} + c \cos \frac{\theta_1}{2} \quad (B.32)$$

$$M25 = N25$$

$$-d_6 \cos (\theta_2 + \theta_3 + \theta_4) \sin \theta_5 + d_5 \sin (\theta_2 + \theta_3 + \theta_4) + a_3 \cos (\theta_2 + \theta_3) + a_2 \cos \theta_2 = x \cos \theta_1 + y \sin \theta_1 \quad (B.33)$$

$$M26 = N26$$

$$-d_6 \cos \theta_5 - d_4 = y \cos \theta_1 - x \sin \theta_1 \quad (B.34)$$

$$M27 = N27$$

$$-d_6 \sin (\theta_2 + \theta_3 + \theta_4) \sin \theta_5 - d_5 \cos (\theta_2 + \theta_3 + \theta_4) + a_3 \sin (\theta_2 + \theta_3) + a_2 \sin \theta_2 = -d_1 + z \quad (B.35)$$

Produits de transformation de la base au poignet 3 (M1 = N1)

$$M1 = ([M11 + M12i + M13j + M14k], < M15i + M16j + M17k >)$$

$$N1 = ([N11 + N12i + N13j + N14k], < N15i + N16j + N17k >)$$

$$\underline{M11 = N11}$$

$$\text{parce que } \frac{\theta_1 - \theta_5}{2} \quad \text{parce que } \frac{\theta_2 + \theta_3 + \theta_4 + \theta_6}{2} = w \quad (B.36)$$

$$\underline{M12 = N12}$$

$$\text{péché } \frac{\theta_1 + \theta_5}{2} \quad \text{péché } \frac{\theta_2 + \theta_3 + \theta_4 - \theta_6}{2} = \text{un} \quad (B.37)$$

$$\underline{M13 = N13}$$

$$- \cos \frac{\theta_1 + \theta_5}{2} \quad \text{péché } \frac{\theta_2 + \theta_3 + \theta_4 + \theta_6}{2} = b \quad (B.38)$$

$$\underline{M14 = N14}$$

$$- \text{péché } \frac{\theta_5 - \theta_1}{2} \quad \text{parce que } \frac{\theta_2 + \theta_3 + \theta_4 - \theta_6}{2} = c \quad (B.39)$$

$$\underline{M15 = N15}$$

$$\begin{aligned} -d_6 \cos \theta_1 \cos (\theta_2 + \theta_3 + \theta_4) \sin \theta_5 + d_6 \cos \theta_5 \sin \theta_1 + d_4 \sin \theta_1 + d_5 \\ \cos \theta_1 \sin (\theta_2 + \theta_3 + \theta_4) + a_3 \cos \theta_1 \cos (\theta_2 + \theta_3) \\ + a_2 \cos \theta_1 \cos \theta_2 = x \end{aligned} \quad (B.40)$$

$$\underline{M16 = N16}$$

$$\begin{aligned} -d_6 \cos \theta_1 \cos \theta_5 - d_4 \cos \theta_1 + d_6 \cos (\theta_2 + \theta_3 + \theta_4) \sin \theta_1 \sin \theta_5 \\ + d_5 \sin (\theta_2 + \theta_3 + \theta_4) + a_3 \cos (\theta_2 + \theta_3) + a_2 \cos \theta_2 = y \end{aligned} \quad (B.41)$$

$$\underline{M17 = N17}$$

$$\begin{aligned} -d_6 \sin (\theta_2 + \theta_3 + \theta_4) \sin \theta_5 - d_5 \cos (\theta_2 + \theta_3 + \theta_4) + a_3 \sin (\theta_2 + \theta_3) \\ + a_2 \sin \theta_2 + d_1 = z \end{aligned} \quad (B.42)$$

Annexe C

Produits de transformation pour Épaule au poignet 1

Cette partie montre les produits de transformation pour la structure cinématique du robot UR3 à partir du cadre 2 (épaule) au cadre 4 (poignet 1).

Nous définissons le produit de quaternion dual M_i comme :

$$M_i = Q_i \quad M_{i+1} \text{ où } 2 \leq i \leq 4$$

$$M_4 = Q_4$$

$$M_3 = Q_3 \quad M_4 = Q_3 \quad Q_4$$

$$M_2 = Q_2 \quad M_3 = Q_2 \quad Q_3 \quad Q_4$$

Nous définissons le produit quaternionique N_i comme suit :

$$N_{i+1} = Q^{-1}_i \quad N_i \text{ où } 2 \leq i \leq 4$$

$$N_2 = Q_2 = 4[w_4, < a_4, b_4, c_4 >, < x_4, y_4, z_4 >]$$

$$N_3 = Q^{-1}_2 \quad N_2 = Q^{-1}_2 \quad Q_2 \quad 4$$

$$N_4 = Q^{-1}_3 \quad N_3 = Q^{-1}_3 \quad Q^{-1}_3 \quad Q_2 \quad 4$$

Annexe C. Produits de transformation de l'épaule au poignet 1

Produits de transformation de l'épaule au poignet 1 (M4 = N4)

$$M4 = ([M41 + M42i + M43j + M44k], < M45i + M46j + M47k >)$$

$$N4 = ([N41 + N42i + N43j + N44k], < N45i + N46j + N47k >)$$

$$\underline{M41 = N41}$$

$$\text{parce que } \frac{\theta 4}{2} = w4 \cos \frac{\theta 2 + \theta 3}{2} - b4 \text{ péché } \frac{\theta 2 + \theta 3}{2} \quad (C.1)$$

$$\underline{M42 = N42}$$

$$0 = a4 \cos \frac{\theta 2 + \theta 3}{2} + c4 \text{ péché } \frac{\theta 2 + \theta 3}{2} \quad (C.2)$$

$$\underline{M43 = N43}$$

$$- \text{péché } \frac{\theta 4}{2} = w4 \text{ péché } \frac{\theta 2 + \theta 3}{2} + b4 \cos \frac{\theta 2 + \theta 3}{2} \quad (C.3)$$

$$\underline{M44 = N44}$$

$$0 = -a4 \text{ péché } \frac{\theta 2 + \theta 3}{2} + \cos c4 \frac{\theta 2 + \theta 3}{2} \quad (C.4)$$

$$\underline{M45 = N45}$$

$$0 = -a2 \cos \theta 3 - a3 + x4 \cos (\theta 2 + \theta 3) + z4 \sin (\theta 2 + \theta 3) \quad (C.5)$$

$$\underline{M46 = N46}$$

$$0 = y4 + d4 \quad (C.6)$$

$$\underline{M47 = N47}$$

$$0 = a2 \sin \theta 3 - x4 \sin (\theta 2 + \theta 3) + z4 \cos (\theta 2 + \theta 3) \quad (C.7)$$

Produits de transformation de l'épaule au poignet 1 (M3 = N3)

$$M3 = ([M31 + M32i + M33j + M34k], < M35i + M36j + M37k >)$$

$$N3 = ([N31 + N32i + N33j + N34k], < N35i + N36j + N37k >)$$

$$\underline{M31 = N31}$$

$$\text{parce que } \frac{\theta_3 + \theta_4}{2} = w_4 \cos \frac{\theta_2}{2} - b_4 \text{ péché } \frac{\theta_2}{2} \quad (C.8)$$

$$\underline{M32 = N32}$$

$$0 = a_4 \cos \frac{\theta_2}{2} + c_4 \text{ péché } \frac{\theta_2}{2} \quad (C.9)$$

$$\underline{M33 = N33}$$

$$- \text{péché } \frac{\theta_3 + \theta_4}{2} = w_4 \text{ péché } \frac{\theta_2}{2} + b_4 \cos \frac{\theta_2}{2} \quad (C.10)$$

$$\underline{M34 = N34}$$

$$0 = -a_4 \sin \frac{\theta_2}{2} + \cos c_4 \frac{\theta_2}{2} \quad (C.11)$$

$$\underline{M35 = N35}$$

$$a_3 \cos \theta_3 = -a_2 + x_4 \cos \theta_2 + z_4 \sin \theta_2 \quad (C.12)$$

$$\underline{M36 = N36}$$

$$-d_4 = y_4 \quad (C.13)$$

$$\underline{M37 = N37}$$

$$a_3 \sin \theta_3 = -x_4 \sin \theta_2 + z_4 \cos \theta_2 \quad (C.14)$$

Annexe C. Produits de transformation de l'épaule au poignet 1

Produits de transformation de l'épaule au poignet 1 ($M2 = N2$)

$$M2 = ([M21 + M22i + M23j + M24k], < M25i + M26j + M27k >)$$

$$N2 = ([N21 + N22i + N23j + N24k], < N25i + N26j + N27k >)$$

$$\underline{M21 = N21}$$

$$\text{parce que } \frac{\theta_2 + \theta_3 + \theta_4}{2} = w_4 \quad (C.15)$$

$$\underline{M22 = N22}$$

$$0 = a_4 \quad (C.16)$$

$$\underline{M23 = N23}$$

$$\text{- péché } \frac{\theta_2 + \theta_3 + \theta_4}{2} = b_4 \quad (C.17)$$

$$\underline{M24 = N24}$$

$$0 = c_4 \quad (C.18)$$

$$\underline{M25 = N25}$$

$$a_3 \cos(\theta_2 + \theta_3) + a_2 \cos \theta_2 = x_4 \quad (C.19)$$

$$\underline{M26 = N26}$$

$$-d_4 = y_4 \quad (C.20)$$

$$\underline{M27 = N27}$$

$$a_3 \sin(\theta_2 + \theta_3) + a_2 \sin \theta_2 = z_4 \quad (C.21)$$

Annexe D

Liste des tableaux et figures

Liste des tableaux

3.1 Tableau de multiplication des quaternions	27
4.1 Paramètres UR3 Denavit-Hartenberg	43
4.2 Paramètres Denavit-Hartenberg pour UR3 de [20]	57

Liste des figures

2.1 Représentation vectorielle de $z = a + ib$ avec $\text{Re}z = a$ et $\text{Im}z = b$.	17
2.2 Représentation en coordonnées polaires du nombre complexe $z = a + ib$ de longueur r et d'angle θ .	17
2.3 Conjugaison complexe.	18
2.4 Addition des nombres complexes $3 + 2i$ et $1 + 3i$.	20
2.5 Multiplication de nombres complexes z_1 et z_2 .	21
2.6 Multiplication de nombres complexes -1 et $-i$.	21
2.7 Multiplication de wz où $z = i$.	22
3.1 Un portrait de Sir William Rowan Hamilton ¹ .	24
3.2 Plaque commémorative de la découverte des quaternions. ²	24
3.3 Diagramme de multiplication de quaternions.	27
3.4 Rotation du vecteur $(2, 1, 1)$ de [22].	35
4.1 Robot Universel UR3.	41
4.2 Structure cinématique de UR3 en position zéro ($\theta_1, 2, 3, 4, 5, 6 = 0$) [3].	42
4.3 Repère de coordonnées de UR3 ($\theta_1, 2, 3, 4, 5, 6 = 0$) de [11].	47
4.4 Structure du robot UR3 du cadre 1 au cadre 5 (de [3]).	48
4.5 Translation du cadre 6 (poignet 3) au cadre 5 (poignet 2) [3].	49
4.6 Manipulateur planaire 3R formé par les articulations UR3 2 à 4 de [3].	53
4.7 Paramètres DH UR3 Diagramme de [20].	57

Bibliographie

- [1] Robotique : Cinématique et fondements mathématiques.
<https://www.edx.org/course/robotics-kinematics-mathematical-pennx-robo1x>.
- [2] M.Ahr. L'héritage des robots universels de Rossum.
<https://www.denofgeek.com/us/culture/279175/the-legacy-of-rossums-universal-robots>, 2019.
- [3] R. Andersen. Cinématique d'un UR5. Rapport technique, Université d'Aalborg
sit  , 2018.
- [4] Y. Aydin et S. Kucuk. Cinématique inverse basée sur les quaternions pour les
robots manipulateurs industriels avec poignet d'Euler. 2006 Internationale IEEE
Conf  rence sur la m  catronique, ICM, 2006.
- [5] J.Barbic. Quaternions et rotations. Rapport technique, Universit   de
Californie du Sud, 2011.
- [6] M. Ben-Ari. Un tutoriel sur les angles d'Euler et les quaternions. 2018.
- [7] Z. Chen et J. Hung. Application de Quaternion dans le contr  le de robot.
10e Congr  s mondial triennal de l'IFAC, 1987.
- [8] J. Diebel. Repr  senter l'attitude : les angles d'Euler, les quaternions unitaires et
Vecteurs de rotation. Rapport technique, Universit   de Stanford, 2006.
- [9] PR Evans. Rotations et matrices de rotation. Rapport technique, MRC
Laboratoire de biologie mol  culaire, 2001.
- [10] WR Hamilton. El  ments de Quaternions. La presse de l'Universite de Cambridge,
1866.
- [11] K. Hawkins. Cinématique inverse analytique pour Universal Robots UR
5/UR-10 Bras. Rapport technique, 2013.

Bibliographie

- [12] AO Colline. Cinématique : pourquoi les robots se déplacent comme ils le font. [https://blog.robotiq.com/kinematics-why-robots-move-like-they do](https://blog.robotiq.com/kinematics-why-robots-move-like-they-do), 2015.
- [13] J. Huerta. Présentation des Quaternions. Rapport technique, Fullerton College, 2010.
- [14] B. Kenwright. Un guide pour débutants sur les doubles quaternions : ce qu'ils sont, comment ils fonctionnent et comment les utiliser pour la 3D. Rapport technique, Université de Newcastle, Royaume-Uni.
- [15] S. Kucuk et Z. Bingul. Cinématique du robot : cinématique directe et inverse. Rapport technique, Université Kocaeli, 2006.
- [16] S. LaValle. Algorithmes de planification - La matrice de transformation homogène. <http://planning.cs.uiuc.edu/node111.html>, 2006.
- [17] A. Larios. Rotations et quaternions. Rapport technique, Stanford Université, 1995.
- [18] JE Marsden et MJ Hoffman. Analyse complexe de base. WHFreeman & Co Ltd, New York, États-Unis, 3 édition, 1999.
- [19] Mataric. L'abécédaire de la robotique. Massachusetts Institute of Technology, Londres, Angleterre, 2007.
- [20] U. Robots. Paramètres pour les calculs de cinématique et de dynamique. <https://www.universal-robots.com/how-to-and-faq/faq/ur-faq/parameters-for-calculations-of-kinematics-and-dynamics-45257/>.
- [21] B. Siciliano et O. Khatib. Manuel Springer de robotique. Springer, Berlin, 2e édition, 2016.
- [22] M. Sunardi. Cinématique 3D. Rapport technique, 2006.
- [23] Tekkotsu. Cinématique. <http://www.tekkotsu.org/Kinematics.html>, 2010.
- [24] J. Voight. Algèbres des quaternions, volume 0.9.14. 2018.
- [25] A. Watt et M. Watt. Techniques avancées d'animation et de rendu. ACM Press, New York, États-Unis, 1992.